UNIVERSITY OF AGDER

DOCTORAL DISSERTATIONS AT THE UNIVERSITY OF AGDER 208

AN ACCESS CONTROL MODEL TO FACILITATE HEALTHCARE INFORMATION ACCESS IN CONTEXT OF TEAM COLLABORATION

Mohamed Ali Saleh Abomhara

Mohamed Ali Saleh Abomhara

An Access Control Model to Facilitate Healthcare Information Access in Context of Team Collaboration

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at
the Faculty of Engineering and Science, Specialization in
Information and Communication Technology

University of Agder
Faculty of Engineering and Science
2018

Doctoral Dissertations at the University of Agder 208
ISBN: 978-82-7117-907-6
ISSN: 1504-9272

Printed by Wittusen & Jensen
Oslo

Dedicated to the memory of my father

Ali Saleh Abomhara

Separated by death, together by love

(1944-2013)

Abstract

The delivery of healthcare relies on the sharing of patients information among a group of healthcare professionals (so-called multidisciplinary teams (MDTs)). At present, electronic health records (EHRs) are widely utilized system to create, manage and share patient healthcare information among MDTs. While it is necessary to provide healthcare professionals with privileges to access patient health information, providing too many privileges may backfire when healthcare professionals accidentally or intentionally abuse their privileges. Hence, finding a middle ground, where the necessary privileges are provided and malicious usage are avoided, is necessary. This thesis highlights the access control matters in collaborative healthcare domain. Focus is mainly on the collaborative activities that are best accomplished by organized MDTs within or among healthcare organizations with an objective of accomplishing a specific task (patient treatment).

Initially, we investigate the importance and challenges of effective MDTs treatment, the sharing of patient healthcare records in healthcare delivery, patient data confidentiality and the need for flexible access of the MDTs corresponding to the requirements to fulfill their duties. Also, we discuss access control requirements in the collaborative environment with respect to EHRs and usage scenario of MDTs collaboration. Additionally, we provide summary of existing access control models along with their pros and cons pertaining to collaborative health systems.

Second, we present a detailed description of the proposed access control model. In this model, the MDTs is classified based on *Belbin*'s team role theory to ensure that privileges are provided to the actual needs of healthcare professionals and to guarantee confidentiality as well as protect the privacy of sensitive patient information. Finally, evaluation indicates that our access control model has a number of advantages including flexibility in terms of permission management, since roles and team roles can be updated without updating privilege for every user. Moreover, the level of fine-grained control of access to patient EHRs that can be authorized to healthcare providers is managed and controlled based on the job required to meet the *minimum necessary* standard and *need-to-know* principle. Additionally, the model does not add significant administrative and performance overhead.

Acknowledgments

First and foremost, I would like to thank my advisors, Prof. *Geir M. Køien* and Prof. *Vladimir A. Oleshchuk* for their guidelines, encouragement, support and confidence in me. I appreciate all their contributions of time, ideas, and professional suggestions to make my PhD experience productive and stimulating. I sincerely thank Assoc.prof. Lillian Røstad and Prof. Tuomas Aura for their comments which were highly insightful and invaluable in improving this thesis.

I would also like to thank all my colleagues and friends at University of Agder, with whom I have enjoyed working and who have helped me with courses and research. Heartfelt thanks to *Mehdi ben Lazerg* for his endless support and assistance on the rough road toward completing this thesis. Special thanks to *Basel Kikhia*, *Huihui Yang*, *Indika Balapuwaduge*, *Lakshmikanth Guntupalli*, *Vimala Nunavath*, *Mohamed Hamid*, *Martin Gerdes*, *Berglind Smaradottir*, *Maurice Isabwe*, *Hossein Baharmand*, *Meisam Naderi*, *Mohammed Talab*, *Alireza Borhani*, *Rym Hicheri*, *Leila Ben Saad*, *Mohamed Elnourani*, *Ahmed Aboughonim* and many others for their friendship and sociability. I am very grateful to *Ali Abdulwahab* and *Rita Solvik-Nilssen* (close friends in Grimstad) for their great hospitality. They have treated me like family, which has really helped to ease my homesickness.

In addition, I would like to express my appreciations for all the help, support and assistance I have received from the administration and the entire Faculty of Engineering at University of Agder, especially *Tonje Sti*, *Emma Elizabeth Horneman* and *Kristine Evensen Reinfjord*. Thanks also to Research School of Computer and Information Security (COINS) for funding of travel costs for some of Ph.D. courses and Ph.D. student seminars.

Last but definitely not least, my deepest gratitude and most heartfelt thanks to my family for their endless love and support. Without their cooperation, this thesis would have never existed. I am deeply indebted to my mother, brothers, and sisters who have been very supportive and encouraging of everything I have undertaken.

Mohamed Abomhara
November 2018
Grimstad, Norway

List of Publications

The outcome of this study is mainly over-viewed by a number of international conferences and journals publications where the author of this dissertation has the major contribution. The following list gives an overview of the published papers.

Paper A: **Mohamed Abomhara**, Berglind Smaradottir, Køien Geir, and Martin Gerdes "Sharing With Care: Multidisciplinary Teams and Secure Access to Electronic Health Records", *Proceedings of the* 11^{th} *International Joint Conference on Biomedical Engineering Systems and Technologies* , Vol 5: HEALTHINF 2018, ISBN 978-989-758-281-3, pages 379-386.

Paper B: **Mohamed Abomhara**, Køien Geir M, Vladimir A. Oleshchuk, and Mohamed Hamid, "Towards Risk-Aware Access Control Framework for Healthcare Information Sharing", *Proceedings of the* 4^{th} *International Conference on Information Systems Security and Privacy*, Volume 1: ICISSP 2018, ISBN 978-989-758-282-0, pages 312-321.

Paper C: **Mohamed Abomhara**, Huihui Yang, Køien Geir M, and Mehdi Ben Lazreg, "Work-based Access Control Model for Cooperative Healthcare Environments: Formal Specification and Verification", *Journal of Healthcare Informatics Research*, vol 1 no 1, pages 19–51, 2017.

Paper D: **Mohamed Abomhara** and Køien Geir M, "Towards an access control model for collaborative healthcare systems", *Proceedings of the* 9^{th} *International Joint Conference on Biomedical Engineering Systems and Technologies*, Vol 5: HEALTHINF 2016, ISBN 978-989-758-170-0, pages 213-222.

Paper E: **Mohamed Abomhara**, Huihui Yang, and Køien Geir M, "Access control model for cooperative healthcare environments: Modeling and verification", *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 46–54, 2016.

Paper F: **Mohamed Abomhara** and Mehdi Ben Lazreg, "UML/OCL-based modeling of work-based access control policies for collaborative healthcare systems", 18^{th} *IEEE International Conference on e-Health Networking, Applications and Services 2016 (Healthcom)*, pages. 1–6, 2016.

Paper G: **Mohamed Abomhara** and Henrik Nergaard, "Modeling of Work-Based Access Control for Cooperative Healthcare Systems with XACML", *The* 5^{th} *International Conference on Global Health Challenges (GLOBAL HEALTH 2016)*, ISBN: 978-1-61208-511-1, pages 14–21, 2016.

Paper H: **Mohamed Abomhara** and Huihui Yang, "Collaborative and Secure Sharing of Healthcare Records Using Attribute-Based Authenticated Access", *International Journal on Advances in Security*, vol 9 no 3 & 4, pages 184–195, 2016.

Paper I: **Mohamed Abomhara** and Huihui Yang, "Attribute-based authenticated access for secure sharing of healthcare records in collaborative environments", *Proceedings of the The* 8^{th} *International Conference on eHealth, telemedicine, and social medicine 2016 (eTELEMED)*, ISBN: 978-1-61208-470-1, pages. 1–7.

Paper J: **Mohamed Abomhara**, Martin Gerdes, and Køien Geir M, "A STRIDE-Based Threat Model for Telehealth Systems", *Norsk informasjonssikkerhetskonferanse (NISK)*,vol 8 no 1, pages. 82–96, 2015.

Paper K: **Mohamed Abomhara** and Køien Geir M, "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks", *Journal of Cyber Security*, vol 4 no 1, pages. 65–88, 2015.

Contents

List of Figures

List of Listings

List of Tables

List of Abbreviations

ABA	Attribute-Based Authentication
ABAC	Attribute-Based Access Control
ACL	Access Control list
ACPT	Access Control Policy Testing
BLAC	Bilayer Access Control
C-TMAC	Contexts-Based Team-Based Access Control
DAC	Discretionary Access Control
EHR	Electronic Health Records
EMF	Eclipse Modeling Framework
EPR	Electronic Patient Record system
GB-RBAC	Group-based Role-Based Access Control
GDPR	General Data Protection Regulation
GP	General Practitioner
HIPPA	Health Insurance Portability and Accountability Act
HIMSS	Healthcare Information and Management Systems Society
HL7	Health Level Seven
ITRC	Identity Theft Resource Centre
MAC	Mandatory Access Control
MDT	Multidisciplinary Team
MLS	Multi-Level Security
NHN	Norwegian Health Network
NHS CRS	National Health Service's Care Record Service
NIST	National Institute of Standards and Technology
OCL	Object Constraints Language
OMG	Object Management Group
PBAC	Purpose Based Access Control
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point

PHI	Protected Health Information
PIP	Policy Information Point
PRP	Policy Retrieval Point
PSCA	Privacy and Security Conceptual Architecture
RBAC	Role-Based Access Control
SCR	Summary Care Record
STD	Sexually Transmitted Disease
SoD	Separation of Duty
TBAC	Task-Based Access Control
TMAC	Team-Based Access Control
TT-RBAC	Team and Task based Role-Based Access Control
UML	Unified Modeling Language
WBAC	Work Based Access Control
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
YAWL	Yet Another Workflow Language

Chapter 1

Introduction

This chapter presents the background and motivation of this PhD research. Section 1.1 provides the background knowledge; Section 1.2 discusses the motivation for this research; Section 1.3 highlights the research questions; Section 1.4 explains the research method and contributions; Section 1.5 presents the research scope; finally, Section 1.6 outlines this thesis organization.

1.1 Background Knowledge

Multidisciplinary teams (MDTs) [112, 131] and electronic health records (EHRs) [161] have become a vital part of modern healthcare delivery [34, 80, 339]. Daily clinical care necessitates the collaborative support of MDTs, including healthcare professionals (e.g., physicians and nurses) and healthcare organizations (e.g., clinics and hospitals). Moreover, healthcare providers (Figure 1.1) and patients employ EHRs widely to create, manage and share health information efficiently and effectively [1, 19, 375]. The benefits of using EHRs include allowing patients to access their own information through patient portals and allowing healthcare providers to access and share patient information more easily [245, 335]. However, health records digitization causes greater abuse and misuse potential against patients and healthcare providers alike [124, 274, 333].

The following subsections provide an overview of MDTs' work and EHRs in healthcare services, followed by security and legal challenges to EHR solutions. A summary of existing access control models along with their pros and cons pertaining to collaborative health systems concludes the section.

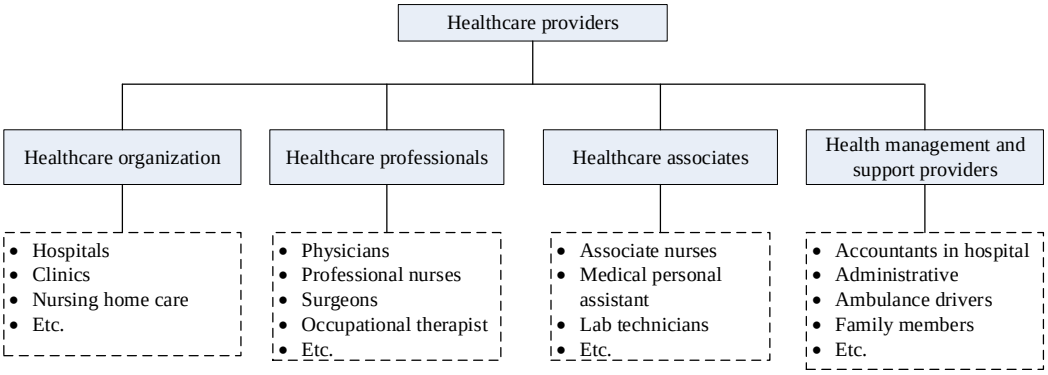


Figure 1.1: Health workforce classification

1.1.1 Multidisciplinary Team Collaboration

The *World Health Organization* (WHO) [137] defines healthcare providers as "an individual healthcare professional, a group or an organization that delivers care services to individuals or communities for the purpose of promoting, maintaining, monitoring or restoring health" [98, 393]. According to WHO classification, a group (also so-called healthcare team or MDT) comprising a variety of professionals and associate professionals as well as health management and support providers, who are involved in providing coordinated and comprehensive care. Figure 1.1 shows an example of healthcare providers classification. Moreover, WHO allows healthcare teams to be distinguished based on the degree of interaction among members and the sharing of responsibility for care.

A multidisciplinary team (MDT) is defined as a group of healthcare professionals from different disciplines, who ideally possess a variety of skills necessary to provide specific patient services [106, 131, 195]. The main aims of MDTs are to deliver effective patient care and improve the outcomes of patients with complex chronic diseases, such as diabetes, cancer, and heart disease [112, 132]. Several studies highlight the importance and effectiveness of MDTs [62, 135, 195, 207, 249, 353, 394]. A typical example of patient care involving an MDT is a pregnant woman (*Sana*) with diabetes who develops a pulmonary embolism (PE) [254]. Her medical care team may include (but is not limited to) an obstetrician, an endocrinologist, a respiratory physician, nurses and others.

Despite the many advantages of MDTs, their success can be affected by several challenges and barriers [246, 281, 384]: e.g., insufficient organization and resource management, poor coordination and communication, as well as resource security and privacy violation [92, 131]. If MDT efforts are not managed and organized properly, the productivity may suffer. Good coordination and communication skills

are at the core of patient safety [208, 281]. When healthcare providers engage in an MDT activity, they are required to switch between varying tasks and roles of distinct nature [246]. Hence, the MDT environment ought to include systems such as electronic health records (EHRs) to assist with task switching accordingly [92]. Such systems facilitate good resource (e.g., patient's EHRs) communication between the MDT and the patient, as well as ensure the availability, confidentiality and integrity of resources by providing them only to MDT members with suitable authorization [59].

1.1.2 Electronic Health Records (EHRs)

Electronic health (*e-Health*) refers to the use of information and communication technologies (ICT) in healthcare services [119]. Governments have introduced broad e-Health reforms (e.g., the Norwegian coordination reform [266, 267], the European Commission's eHealth Action Plan 2012-2020 [70, 115, 117, 118, 350] and the American Recovery and Reinvestment Act (ARRA 2009) [58, 285, 347]) to encourage e-Health technology adoption by promoting the meaningful use of EHR solutions, among other provisions. The aims include enhancing healthcare quality, facilitating easy collaboration and interaction between patients and healthcare providers, as well as supporting close cooperation between healthcare professionals from different organizations.

An EHR is a compilation of various types of patient health records that are stored in electronic format [120]. EHR integration in healthcare organizations offers potential benefits in terms of improved care quality [80, 203, 335], simplified management, and efficient in-patient and out-patient health record exchanges [245]. Thus, it is possible to reduce costs associated with patient care and administrative overhead [33, 188, 203]. For example, the openEHR Foundation [87, 199, 273] started a project to develop an open and more comprehensive componentized architecture that includes a secure approach to health data sharing in a distributed environment. Several countries (e.g., Norway, the United Kingdom (UK) and Canada) have established EHR strategies that involve openEHR [63, 66, 87]. Further examples of organization-based projects include a collaborative telemedicine system for remote chronic obstructive pulmonary disease (COPD) monitoring, developed within the European Union (EU) project *United4Health* [365] to support healthcare service collaboration across healthcare organizations in the Norwegian southeast health region [338]. This system allows both hospitals and municipal healthcare services access to patient information [339, 340]. Chapter 2 provides more examples of EHR initiatives in Norway and other countries.

EHRs can assist overcoming traditional MDT barriers by enabling communication among participants and providing rapid access to health records when distance is involved [147, 371, 392]. EHRs improve the MDT work flow and enable more seamless collaboration and information exchanges between healthcare professionals within and among healthcare organizations [34, 80, 81, 147, 282, 299]. Both healthcare providers and patients can benefit from the EHR management and sharing features. One healthcare professional can create and instantly share patient records while other professionals can review and extend the records digitally. For example, high-resolution collaborative medical imaging sharing systems [26, 287]. Such systems provide medical imaging repositories for physicians to diagnose and treat particular diseases effectively as a team.

Even though EHR systems may improve healthcare quality, significant related barriers remain (e.g., cost, technical issues, legal considerations, security and privacy issues) [9, 13, 41, 91, 124, 167, 214, 274, 349, 369]. The focus of this research is on access control as well as proper use and sharing of patient health information. A major concern is to prevent (1) privileged healthcare professionals from disclosing sensitive health information improperly and (2) persons who can take advantage of the MDT environment from having unauthorized access to health information [121, 122, 124, 133, 244, 284, 333, 405]. Improper disclosure or unauthorized access may occur when someone within the MDT accesses shared resources for unethical reasons (insider threat [68, 178, 284]), for instance accessing a patient's private information for personal gain (more about insider threats in Chapter 2).

1.1.3 Security Challenges and Legal Requirements of EHRs

Security and privacy are major concerns for patients and healthcare providers worldwide [124, 231, 291]. Patient health records are regarded as private, because they may contain sensitive personal details. There is even greater sensitivity about more serious medical conditions, often; due to fear or shame (e.g., lung cancer, sexually transmitted diseases); or because of possible social embarrassment (e.g., mental health problems, being HIV positive) [40]. A study by *Chhanabhai* and *Holt* [86] showed that 73.3% of participants exhibited concern regarding the security and privacy of their health information. *Vodicka et al.* [375] carried out a survey on online access to patient records and found that approximately one-third of participants were concerned about the security and privacy of their health records, particularly regarding who should have access to what health information. Moreover, in 2013, a survey [163] done by *Healthcare Information and Management Systems Society* (*HIMSS*) indicated that two-thirds of respondents had concerns that internal

breaches could compromise electronic information security. An example of internal security breaches at *Howard University Hospital*, *Washington* showed that inadequate data security can affect a large number of people [274]. On May 14^{th}, 2012, one of the hospital's medical technicians was charged with violating her privileges at the hospital to gain access to patients' information (e.g., names, addresses, and diagnosis related information) for a personal gain [319].

Additionally, health IT security firm *Redspin* released an alarming report in 2016 (Redspin's 6^{th} Annual Breach Report: Protected Health Information) [296], which showed that nearly 155 million patient health records had been breached since 2009. An 897% increase in the number of breached patient records was also noted in 2015 compared to 2014. Furthermore, according to a survey by *IBM* and the *Ponemon Institute* in 2017 [342] as well as other reports [183, 232, 368], healthcare data breach costs are the second highest among various industries. Breaches involve health information theft, loss or improper disposal of medical records, and unauthorized access to, or disclosure of health information. The above mentioned findings demonstrate that it is essential to address security and privacy concerns regarding EHRs before patients and healthcare providers can fully accept EHRs.

Not only do patients and healthcare providers demand security and privacy protection for patient health records, but in most countries, the law requires this as well. Standards and legislation have defined access restrictions to protect patient privacy and means of processing patient health records securely. *Health Level Seven* (HL7) [168] is a standard for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice along with health service management, delivery and evaluation. The main requirement for protecting privacy in HL7 and other regulations [333] is that health data sharing must be controlled by patient consent while allowing differential access to aspects of the health information depending on the sensitivity of the information as perceived by the patient. The *Health Insurance Portability and Accountability Act* (HIPAA) [268, 307, 367] is an American legislation to ensure that health information is protected adequately while allowing a health information flow[1] necessary for providing and promoting high-quality healthcare. Europe has similar legislation including *European Union Data Protection Directive* (EU Directive 95/46/EC) [107, 114] provide a comprehensive legal framework for data protection in the EU. The *General Data Protection Regulation* (GDPR), adopted in April 2016, superseded the EU Directive 95/46/EC and enforced on 25^{th} May 2018 [55, 358, 376].

[1]Information flow concerns how the information should proceed to authorized entities, to whom the information should be propagated and what steps and methods should be used to ensure information flow [257].

Over and above, several studies [88, 107, 109, 333, 349, 387, 402] have showed that legislative institutions of most countries (e.g., Norway, the UK, Canada) have ordained laws and policies concerning disclosure and sharing of patient health information. One instance is the *Norwegian Personal Health Data Filing System Act* [88, 265, 387]. The former *Norwegian Personal Health Data Filing System Act* from 2001 prohibited sharing and accessing personal health data across organizations [88]. Each healthcare organization was obliged to have its own internal EHR system to which only the institution's own employees could legally be granted access. This restriction was altered in 2015 when the *Norwegian code of conduct for information security* [265] and a new *Personal Health Data Filing System Act* were passed and replaced the law from 2001. Since then, shared EHRs have been legal [88]. The goal is to facilitate cooperation and increase the quality of medical treatment and care when such care involves more than one health provider. More about standards and legislations can be found in [107, 109, 333].

Accordingly, such standards and legislation also provide security and privacy suggestions to address the need to protect health information. Access control is critical to helping manage problems related to unauthorized and improper access [124]. For example, a specialist may only access information of patients he/she is treating. In overcoming authorization and improper access issues associated with EHRs, some access control models have been proposed: e.g., role-based access control (RBAC) [128], attribute-based access control (ABAC) [173] and others [359, 360].

1.1.4 Access Control Models

Access control is the most common approach to managing information access and controlling legitimate user activities by mediating each user's attempt to access a system resource [104, 374]. The ultimate goal of an access control system is to allocate all users the specific access level necessary to do their job [313]. Since the late 1960s, researchers in the security field have proposed several models to address security challenges related to access control [104]. Discretionary access control (DAC) [202, 314], mandatory access control (MAC) [237, 314], role-based access control (RBAC) [127, 272, 317] and attribute-based access control (ABAC) [173] are examples of access control models.

RBAC is a popular access control model which is widely employed in the health sectors due to the convenience it offers [57, 124, 228, 313]. It is fairly easy to assign users authorization based on their roles. However, RBAC has shortcomings too [222, 306, 313]. For instance, RBAC cannot provide efficient authorization management for collaborations [222]. The main reason being that RBAC focuses on

user permission control according to pre-assigned roles and permission-role assignment relations [227, 228]. In dynamic environments such as MDTs, roles and user-role assignment relations are not fixed during collaborations. Besides, professional roles and users in EHR systems differ in their number and definition. There is also no universal model applicable to role definitions and profiling (i.e., role profiling entails defining the responsibilities) that all health organizations can adopt [369]; thus, authorization management is a significant obstacle to secure access.

Moreover, RBAC is not well-suited for EHRs to handle unplanned and dynamic events (e.g., when a healthcare provider asks other healthcare providers for second opinions) [124, 305, 313]. The *Redspin* report [296] showed that healthcare organizations use multiple means of controlling access to patient information. Two-thirds of respondents reported the usage of at least two access control mechanisms, e.g., user-based and role-based, to control access to data. Furthermore, *Rostad et al.* studied eight systems in Norwegian hospitals and also indicated that most EHR systems employ exception mechanisms in addition to RBAC to handle unplanned and dynamic situations [305]. According to the study, 54% of health records accessed over one month were accessed through an exception mechanism, which overrides the access request denied by RBAC. Another study by *Hystad* and *Fensli* [181] demonstrated that implementing access control in EHR systems in Norway is not sufficiently tailored for treatment processes and exception mechanisms are used extensively. Although exception mechanisms enhance flexibility, their use leads to security and privacy threats [305, 313].

Additionally, RBAC often faces difficulties with enforcing the *need-to-know* principle (i.e., access is only allowed to health information which is relevant to the care process) and the *minimum necessary*[2] standard for disclosing patient records for treatment [6, 22]. Such difficulties are due to problems including, foremost, the requirement that "nothing must interfere with the delivery of care" [313], which implies that it is not possible to simply deny an access request because the predefined policy did not authorize it explicitly. Second, medical records contain a wide range of information and it is infeasible for the policy author (e.g., administrator) to foresee what health information different healthcare providers may need in various situations [378]. Third, healthcare providers cannot decide what information is really necessary in a patient treatment case [313]. Thus, healthcare providers often have unlimited access to patient health records, leading to unaccountable risks to the respective records.

[2]The minimum necessary standard requires the covered entities to evaluate their practices and enhance health information protection as needed to limit unnecessary or inappropriate access to, and the disclosure of protected health information.

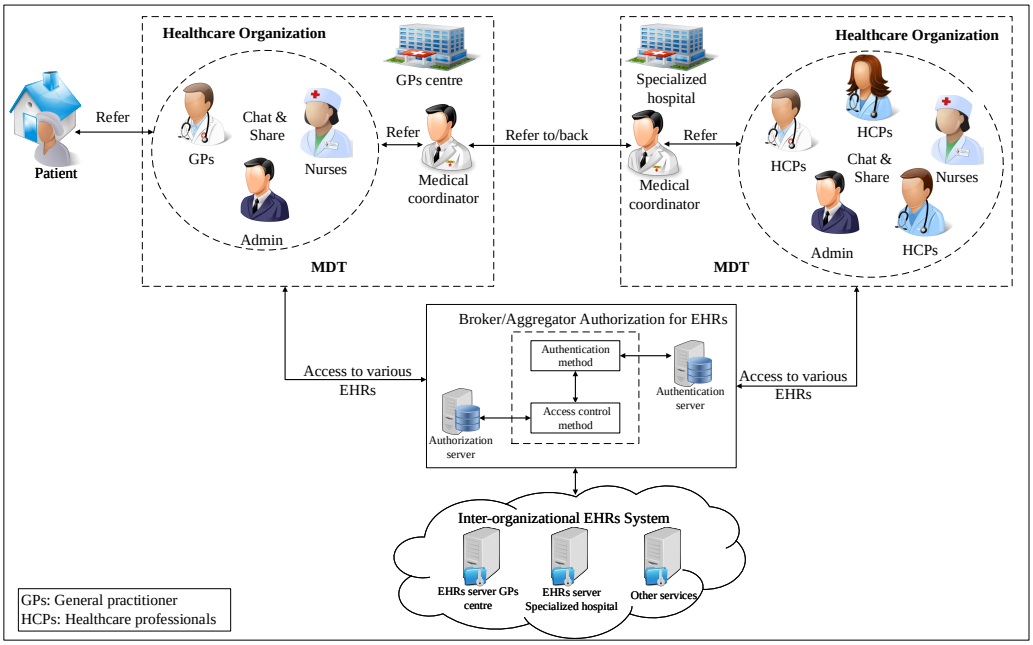


Figure 1.2: EHR scenario where patients and healthcare professionals exchange health information

1.2 Research Motivation

The integrated use of EHRs to enhance healthcare services is promising due to a number of attractive features [245, 335]. One is the improvement in healthcare service quality and delivery by providing healthcare providers access to information they require to provide rapid patient care [38]. Perhaps the most notable contribution of EHRs is the key role in facilitating effective communication and health information sharing between multiple parties (MDTs) to fulfil the information requirements of daily clinical care [80, 282]. As mentioned in Section 1.1.2, one healthcare provider can create and digitally share patient health records (e.g., clinical history, physical examinations, and diagnostic testing results), while other healthcare providers can review the records instantly. Figure 1.2 illustrates the information exchange process in a general EHR scenario, where patients and health professionals as well as health professionals themselves exchange health information.

The EHR exchanges can be implemented in one of two ways: centralized and decentralized (federated) [150, 369, 396]. In centralized health information exchange environment, all health data would be stored in a central repository or database. Healthcare providers would then access that centralized service in order to view patient's health records. In a decentralized environment, each healthcare organization would continue to maintain their own EHRs but the information stored in distributed information system and the health information exchange would act

as a "broker" or pointer service to the location of requested data [351, 389]. Note that the environment described in Figure 1.2 holds no assumption about the EHR location, whether centralized or distributed.

Consider the following clinical case study (adapted from [369]) illustrating what happens from a patient and healthcare provider perspective:

Clinical case study 1: *"A patient (Jones) lives in a town with a large hospital, a small psychiatric institution and several general practitioner (GP) centers. These are all separate organizations with contracts to share relevant information. The patient visits one specific GP regularly who is fully informed about the patient's medical history. Jones has a history of depression that once resulted in a short stay at the psychiatric institution. He is doing well now and his current medication prevents depression relapse. Jones gave the psychiatric institution permission to respond to requests for information only from his GP. The psychiatric institution has sent Jones' discharge information to the GP. Jones also informed his GP that he does not want his psychiatric records disclosed to others, unless disclosure might have serious implications for future treatments. One day, Jones develops a rash and consults his GP who is unsure whether it is an allergic reaction or something else. The GP decides to refer Jones to a dermatologist at the dermatology department at the large hospital in town. The dermatologist wants to know whether there is any medical information regarding allergies and medication about the patient elsewhere. The patient's answers are vague, so the dermatologist decides to ask the GP (with the patient's consent), who responds in compliance with the patient's wishes. The dermatologist also orders a blood test and a skin allergy test at the hospital lab. The lab performs the tests and sends the results to the dermatologist. The skin test reveals a mild allergic reaction to cats. The dermatologist advises Jones to stay away from cats and refers him back to the GP."*

Observations from this clinical case study:

- Several healthcare professionals and organizations have various roles in providing patient care. These include a general practitioner center with GPs and a specialized hospital with specialists.
- The healthcare professionals are organized in dynamic teams. For example, when the GP requests a consultation with the dermatology department, a team of specialists (dermatologists) forms in response. The team can comprise of a single or multiple departments (units) within or among the healthcare organizations. For example, when the dermatologist requests a blood test, the medical laboratory in another department can do the test.

- Every participant needs to obtain the requested medical records for treatment on a *need-to-know* basis (i.e., during patient treatment only) and *minimum necessary* standard (i.e., only health information related to the current patient case) [6]. For instance, if the supporting party is included solely for consultation purposes (i.e., consultation in the treatment) regarding the disease, only information essential for diagnosis should be provided.

- Patients must be confident that their sensitive health information is secured against unauthorized disclosure and is only available to authorized healthcare professionals involved in the patient's treatment. The patients may also need to be able to conceal certain information from certain team members.

From an access control model and authorization as well as legal frameworks point of view, the following requirements are noted, among many others (discussed in more details in Chapter 2):

1. **Authorized access**: Only MDT members should have a permission (i.e., approval to perform an operation on one or more resources (EHRs)) to patient records. For example, a dermatologist has the right (i.e., ability to take an action) to order a blood test on account of his/her role in the hospital. However, he/she ought to have a permission to order the test only when he/she is a member of the patient's treatment team. Legislation such as *HIPAA Privacy Rules* [307, 366, 367], *Personal Health Data Filing System Act* [387] and *UK Good Medical Practice* [363, 364] stipulates that access to patient health information may only be granted as far as it is necessary for the patient treatment. Furthermore, any access should be in accordance with the rules that apply regarding the duty of confidentiality[3].

2. **Fine-grained access**: All healthcare providers have different responsibilities based on their qualifications such as consultant, associate consultant, principal doctor and residency doctor [407]. Therefore, the healthcare providers' (team members) permissions should reflect their roles in the team. Moreover, permissions should be restricted to specific patient records that are relevant to the current patient case (*minimum necessary* standard). For example, consider *Jones*, who requested that his GP does not disclose his psychiatric records

[3]The duty of confidentiality obliges privacy and respect the confidentiality of the information in the context of privileged communication (e.g., patient-doctor consultations) and medical records is safeguarded. Breach of confidence, inappropriate use or abuse of health records may lead to disciplinary measures [333, 363, 364].

to others unless necessary. According to *HIPAA Privacy Rules* [367], psychotherapy notes should be treated differently from other health information because they are personal notes that contain particularly sensitive information. In this case, only the team members who need this information should have permission to access psychiatric records and not all team members.

3. **Dynamicity**: Such collaborations may dynamically change participants (team members) and trust relationships during the patient treatment. For example, when the dermatologist orders a blood test, a medical technologist would join the treatment team to perform the test. Thus, access control models for MDTs must be dynamic, that is, it should be possible to add or remove participants and also the authorization policies have to explicitly specify which users (team member) from which unit/organizations can access which resources (EHRs).

4. **Audit Logs**: All access permissions to patient records should be logged and the information subject (owner) should be notified. For instance, *Personal Health Data Filing System Act* obligates access to be logged and the information subject has the right to view the logs to find out who has accessed his/her health data [387].

Motivated by the RBAC model shortcomings (Section 1.1.4), to fulfil the mentioned requirements, and in line with previous research [11, 17, 49, 51, 134, 152, 191, 213, 222, 235, 256, 286, 306, 310, 313, 373, 401, 404] on access control models, this research is an effort to address some of challenges in facilitating secure health information exchanges. The focus is on developing an access control model that enables a balance between MDT collaboration and safeguarding sensitive patient information.

1.3 Research Questions

The most pressing concern with deploying access control in a collaborative healthcare environment is deciding on the extent and limits of information sharing. For instance, if the main physician is treating a patient with sensitive information (e.g., consider *Jones*' case and his sensitive psychiatric records), the questions are what information to disclose to an assisting practitioner so that collaboration can be effective and what to conceal to safeguard the patient's privacy.

An analysis of the clinical case study 1 and the main research challenges (Section 1.1) resulted in the following research questions for this study:

RQ 1: *What health information (patient EHRs) should be available and under what circumstances can health information be shared during MDT collaboration?*

RQ 2: *Who should decide on the extent and limits of health information sharing?*

RQ 3: *What are the strengths and weaknesses of existing access control models proposed for healthcare?*

RQ 4: *How can the access control model be extended to support MDT collaboration and health information sharing without adding administrative overhead?*

1.4 Research Method and Research Contributions

This section presents the research method and research contributions.

1.4.1 Research Method

To answer research questions, we adapt the design paradigm [103] which consists of four steps as follows (Figure 1.3):

1. **Build a requirements specification**: In this step, we analyzed the research problem, formulated and defined the access control requirements based on a given clinical case studies (Chapter 1 and 2).

2. **Acquire knowledge**: In this step, we conducted a theoretical study in order to gain a comprehensive understanding of access control models along with their pros and cons with respect to MDT collaboration and EHRs. Books, scientific papers (e.g., conferences, journals and technical reports), strategy reports and policy documents served as various sources of knowledge (Chapter 1 and 2).

3. **Design and implementation**: By considering the requirements and specifications, we proposed an enhanced access control model suitable for collaborative healthcare systems in terms of addressing information sharing and security issues. Chapters 3, 4, 5 and 6 describe the full model.

4. **Evaluation**: We evaluated the proposed model against the requirements and answered the research questions (Chapters 4, 5, 6 and 7).

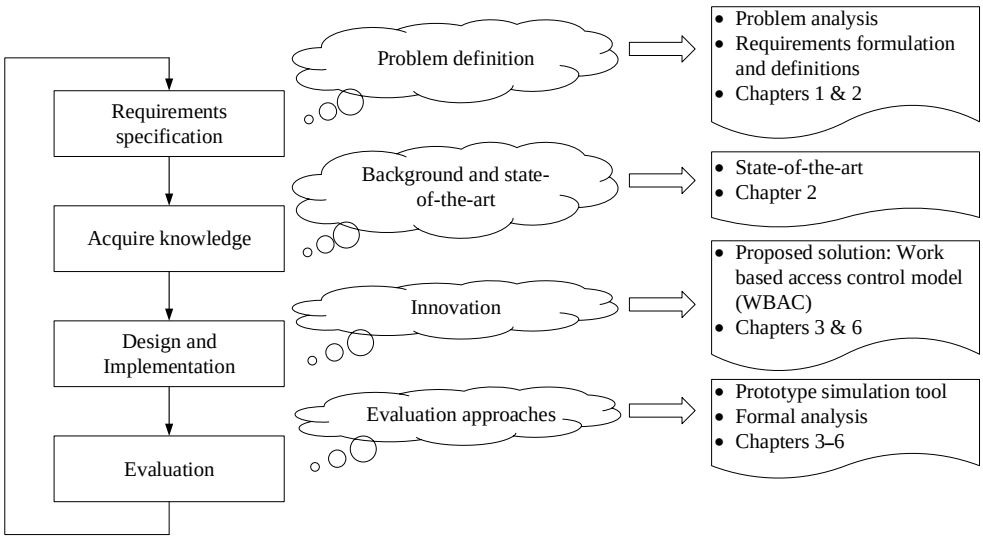


Figure 1.3: Method for this research–main steps

1.4.2 Research Contributions

The main objective is to propose an access control model for healthcare providers. The model provides access permissions so that the appropriate healthcare providers can access patient records and only when they are providing patient care. Moreover, the model does not add significant administrative overhead and is self-administering to a great extent.

The main contributions of this research are as follows:

1. **State-of-the-art**: We investigate and gain a deep understanding of collaborative healthcare environments (MDTs), EHRs and the main insider security issues associated with MDTs and EHRs environments. We also survey existing access control models and present access control requirements in the context of collaborative healthcare environments. This survey could be useful for future investigations in the area of access control and MDTs.

2. **Proposed team role classification:** To address the problem of role definition and profiling, we propose a team role classification based on *Belbin* team role theory [42, 247, 248]. For the purposes of this research, the nine different team roles that *Belbin* identified were rephrased and classified into *thought*, *action* and *management*. Team member must be assigned to one team role based on the goal, task and contributions towards achieving the team's objectives. The team role determines the finer role and the extent of access of each team member. To the best of our knowledge, we are the first to use a team role theory within access control.

3. **Work-based access control:** We propose a work-based access control model (WBAC), which is based on the RBAC and ABAC models. WBAC is extended with the team role concept. Role is used in conjunction with team role to handle access control in dynamic collaborative environments. WBAC is suitable for collaborative healthcare systems in addressing concerns with information sharing and information access. The proposed model ensures that access rights are adapted to the actual needs of healthcare providers. Healthcare providers can access the resources associated with patient treatment but only during the treatment course. Upon treatment completion, access rights should be revoked.

4. **XACML profile for WBAC:** We define a policy structure that we use to express WBAC policies. This structure is based upon using eXtensible Access Control Markup Language (XACML). The experimental results demonstrate the efficiency and scalability of the WBAC approach.

5. **Formal definition and verification**: To evaluate and analyze the security of the proposed model, we first formally define the basic element set and relations in the WBAC model, present the WBAC authorization constraints and define the WBAC access control decision processes. Second, we evaluate the WBAC model to ensure that the security and management requirements of WBAC are met. Moreover, a generic model checking technique, *Access Control Policy testing* (ACPT) [180], is used to verify WBAC policies to ensure that the WBAC policies satisfy the security properties intended by the model. Finally, we evaluate the performance of the proposed model.

6. **Specification and validation of authorization constraints**: We demonstrate how the authorization constraints expressed in the *Object Constraints Language* (OCL) [144, 382] can be implemented, tested and validated using the *Eclipse Modeling Framework* (EMF) [67, 346]. We additionally introduce what objects should be defined in the WBAC model, how the functionality defined in WBAC is arranged into these objects, and how these objects work together to make access control decisions. Finally, we compare WBAC with relevant existing models.

7. **Risk aware access control framework**: We propose a risk assessment framework that facilitates reasoning and managing risk in the WBAC system. Risk-based WBAC makes access decisions by determining the risks associated with access requests and weighing such risks against the risk appetite and

risk tolerance. The WBAC risk assessment framework is flexible and able to handle different risk management scenarios in dynamic environments such as healthcare.

1.5 Limitation of the Research Scope

It is very important to clarify, identify, and describe the limitations of this research scope. Therefore, the scope of this PhD thesis is described as follows:

- This study reflects on experience based on previous studies in the literature.
- The focus of this PhD thesis is restricted to EHR security and privacy challenges. More precisely, the primary focus is on authorized access to health information during MDT collaboration.
- The focus of this PhD thesis is restricted to healthcare professionals and healthcare associates (Figure 1.1) who directly involved in the patient treatment tasks (assessing people, setting goals and making care recommendations, etc.) and require access to patient's EHRs.
- The current study does not highlight any specific EHR system but. It only adapts imaginary but realistic scenarios with focus on health information sharing and team collaboration.

1.6 Thesis Organization

This thesis is organized as follows:

Chapter 2 (State-of-the-Art) addresses relevant work underlying the current research, including an overview of EHR initiatives, collaborative healthcare environments, and insider threats. This is followed by a review of major classical access control models. In addition, existing access control models are compared and a brief discussion is provided on their pros and cons with respect to collaborative healthcare systems.

Chapter 3 (Work Based Access Control (WBAC)) describes the major contributions of this thesis and provides a detailed description of the proposed WBAC model. The main WBAC model components, collaboration work model, proposed team role, resource types and WBAC flow model are described as well. A WBAC policy is subsequently presented using eXtensible Access Control Markup Language (XACML).

Chapter 4 (Formal Definition and Verification of WBAC) presents the formal definition and verification of the proposed model. Definitions of the general WBAC model principles are first given. The basic element set and relations in the WBAC model are formalized, WBAC authorization constraints are presented, and model validity is evaluated using model checking tools. Moreover, the chapter presents a performance analysis of WBAC.

Chapter 5 (Specification and Validation of WBAC Authorization Constraints Using UML and OCL) presents an overview of how UML and OCL are used to specify and analyze the WBAC authorization constraints. Moreover, the chapter presents a comparison of the WBAC and other access control models.

Chapter 6 (Risk Assessment in WBAC Model) describes risk assessment framework for the WBAC model. It provides a summary of basic risk assessment terminology and approaches, followed by a description of the WBAC risk assessment framework and how the WBAC access control model can help mitigate insider risks to minimize the impact of unauthorized access.

Chapter 7 (Conclusions and Future Work) offers a summary the main ideas and the finding of our research. First, it presents certain observations that we have learned from the previous studies. Second, it answers all the research questions. Finally it puts forth proposals for future enhancements followed by conclusions of this thesis.

Chapter 2

State-of-the-Art

This chapter is a review of relevant work related to the present research. Section 2.1 presents an overview of EHR initiatives. Section 2.2 offers a brief summary of collaborative healthcare environment and authorization issues in the healthcare domain. Section 2.3 provides a discussion on access control requirements with respect to EHRs and MDTs work. The chapter also entails a discussion on classical access control models with their strengths and weaknesses regarding collaborative healthcare systems (Section 2.4) along with access control models for collaborative environments (Section 2.5). Section 2.6 describes the research trends in health information access control. Finally, Section 2.7 summarizes the chapter.

2.1 Trends in the EHR Initiatives

An EHR is an electronic health record system used to electronically collect and store information about patient health and care. Healthcare organizations generally own EHRs and healthcare professionals involved in patient care process (e.g., read and write) EHRs [333]. EHR adoption in healthcare has a number of potential benefits as reported in literature [34, 80, 81, 282, 299]. These include, among others, support for activities and processes involved in clinical care delivery within and among healthcare organizations; enhanced patients understanding of their condition through access to their health records; reduced costs, and improved quality of care [33, 188, 245, 335]. EHRs offer efficient means of sharing patient health records among those in healthcare organizations, such as physicians and nurses.

Many countries (e.g., the UK [1], Canada [333], Australia [20] and others [19, 61, 108, 190]) are widely utilizing EHR systems, while several other countries are planning widespread EHR implementation (e.g., Norway [87, 88, 164] and Saudi Arabia [13, 18]) to create, manage and access patient healthcare information efficiently and effectively. The following subsections contain brief discussions of some EHR trends in Canada, the UK and Norway.

2.1.1 The Canadian Healthcare Infoway

The *Canada Health Infoway* is one of many EHR solutions to expand high quality healthcare services across Canada [78, 142]. The initial focus of *Infoway* was to help improve Canadians' healthcare by working with partners to accelerate the development, adoption and effective use of interoperable EHR systems across Canada [19, 410]. It is now aimed at connecting healthcare organizations and encouraging them to produce and share knowledge objects (e.g., patient health records) as well as provide immediate access to patient health information that other healthcare organizations can reuse to support more efficient healthcare delivery, productivity and cost savings.

There are several challenges to *Infoway* implementation [78], the main being with ensuring legal and privacy compliance in health information sharing [333]. The privacy and security conceptual architecture (PSCA) was developed to fill the gap in *Infoway*'s legal as well as security and privacy requirements [74, 75]. PSCA is an attempt to operationalize different jurisdictional consent requirements for care and treatment. In addition, PSCA seeks to ensure the availability of health information, but only to those authorized. Security management within the Canada Health *Infoway* PSCA consists of various facets, including authentication for accurate user identification, access control for controlled access to health information, and secure auditing for secure EHR access logging and use [333].

2.1.2 The UK's National Health Service

In the UK, similar to Canada, an EHR model is evolving at the national level. The *National Health Service's Care Record Service* (NHS CRS) was introduced with the aim to improve healthcare delivery and quality [95, 108, 330]. The original objective of NHS CRS was to ensure that every NHS patient had an individual electronic health record that could be transmitted rapidly between different NHS areas and made available to all NHS healthcare providers anywhere, at all times [108]. At present, only a few electronic records are shared between providers. The electronic

Summary Care Record (SCR) contains limited patient information (prescriptions, allergies and adverse reactions) and is shared between hospitals, GP surgeries, walk-in centres and, from 2017, with community pharmacists [361]. Although healthcare organizations generally own health records and healthcare professionals maintain health records, there is a growing call to empower patients to participate, especially in deciding who has access to their information. The UK government's target is to introduce a comprehensive system of electronic health records in England by 2020 [111, 361]. The intention is that each patient's electronic record will include information about his or her medical history, care preferences and lifestyle (such as diet and exercise). The records should be accessible to all health and social care providers and updated in real-time. Patients should be able to view and annotate a version of their health record online and ultimately have the opportunity to decide on the security and privacy handling of their information at NHS CRS.

According to a briefing paper [277] on patient health records and confidentiality, healthcare providers have the legal right to access patient health records except where the information may cause serious harm to the patient, or would reveal information about another person who has not consented to this disclosure. Therefore, healthcare providers have a duty of confidentiality to patients and must seek their consent before sharing their data [364].

2.1.3 The Norwegian Healthcare System

Over the last years, Norwegian hospitals have concerted great efforts driven by a sequence of strategic plans that the Norwegian directorate for health (Helsedirektoratet [166]) proposed to standardize hospital infrastructure for EHRs [1]. However, interoperable EHR system adaptation in the sector has been low-level due to a number of barriers (e.g., patient involvement as well as legal and ethical matters [387]). According to a survey by *Heimly et al.* [165], almost all hospitals and GPs use a local EHR, or a so-called *Electronic Patient Records* (EPRs) on a daily basis, whereas EHR system use in municipalities (e.g., nursing homes and child health centers) is more limited.

The *Norwegian Health Network* (NHN) is a dedicated secure network that supports communication across healthcare organizations [1]. The main objective of NHN is to connect hospitals, GPs, nursing homes and more recently, pharmacies, to support secure health information exchanges in instances of referrals, requisitions and prescriptions. NHN does not support communication among patients or between patients and healthcare providers; thus, patients do not have access to their health information (unless they request a copy).

A number of ongoing projects with EHR vendors in Norway (e.g., DIPS is the largest EHR vendor in Norway covering 80% of the EHR hospital market and encompassing 80,000 users [87]) aim to develop solutions to give patients access to health information and provide secure health information communication and sharing. Moreover, a number of ongoing national initiatives are aimed at giving inhabitants (Norwegian citizens and foreigners) access to health information. The *Kjernejournal* (Summary Care Records) is an EHR solution that simplifies and speeds up communication between healthcare providers [182]. It contains patient health information entered by GPs and retrieves prescription histories and information from national registries. *Kjernejournal* provides healthcare professionals immediate access to selected, important patient health information residing in national-level data repositories, regardless of where the patient received the treatment. Moreover, patients can access their health records, view the access logs, register new information such as primary contact person and disease history (structured selections), or they may opt out of the records entirely [1]. *Kjernejournal* has been implemented nationally but is still under development. In the future, it could become a part of an interoperable EHR system that empowers patient involvement in healthcare, simplifies the health information exchange between healthcare providers, and hence improves quality of care by offering correct and up-to-date health information.

Whilst literature on *Kjernejournal* implementation is somewhat limited, there are nonetheless a number of commendable studies on the legal and security challenges regarding shared EHR systems and health information protection and privacy [88, 265, 333, 387]. Healthcare organizations are obliged to set up an EHR system that must be organized such that it will satisfy the requirements set in accordance with the law and regulations (cf. *Personal Health Data Filing System Act and Personal Data Act* [265, 387]). The requirements pertain to rules on personal health information disclosure and patient health record access only when necessary in patient treatment.

2.2 Collaborative Healthcare Environment

e-Health collaborative environment is a virtual infrastructure that allows individuals to collaborate with greater ease. It provides the necessary processes and tools to promote teamwork among individuals with similar goals [321]. For example, a team can divide the work and perform it separately (Figure 2.1), thereafter assembling the individual work into a cohesive whole.

Collaboration at healthcare organizations is an integral part of the work pro-

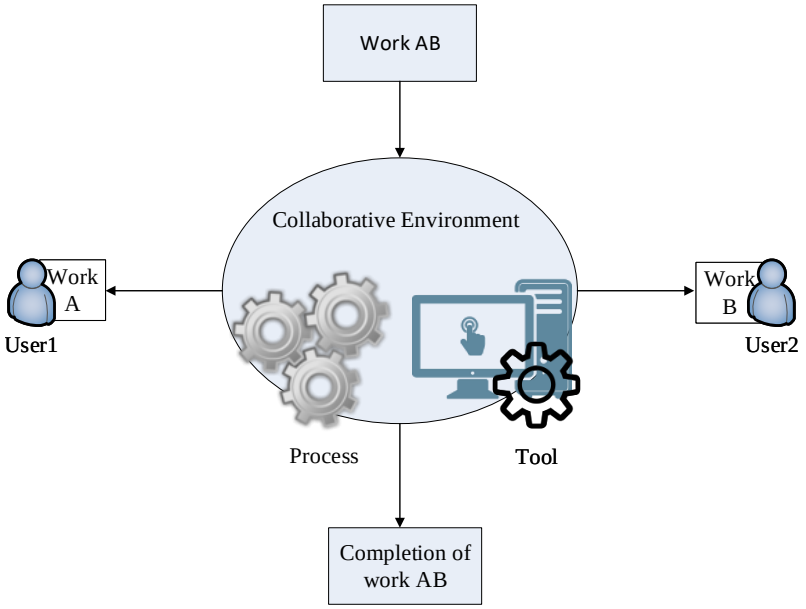
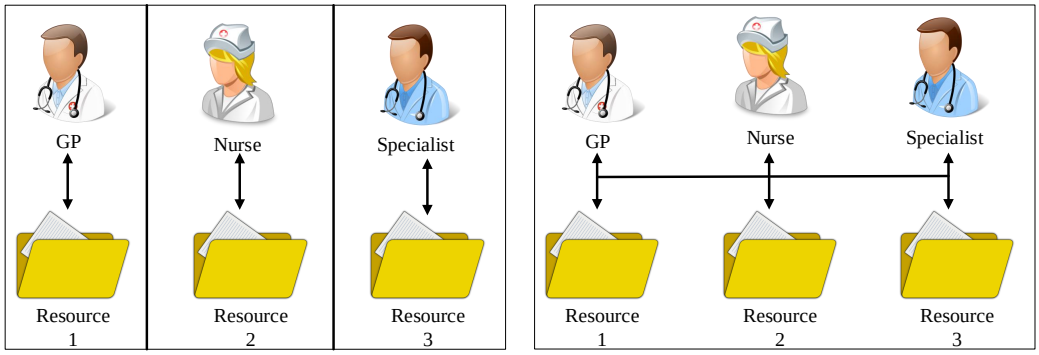


Figure 2.1: Collaborative environment and work sharing

cess, whereby experts with different specializations and backgrounds contribute as a group (MDT) to ensure treatment success [31, 62]. The increasing complexity of the medical domain further amplifies this necessity. As illustrated in the cases of the pregnant woman *Sana* (Section 1.1.1) and patient *Jones* (clinical case study 1), healthcare services typically necessitate collaborative support from multiple parties to fulfill the information requirements of daily clinical care and provide rapid patient care [112, 254]. Healthcare organizations such as hospitals require collaborative support, where patients move among healthcare professionals, laboratories and wards [207]. Collaboration among healthcare organizations is also essential for patients been transferred from one healthcare provider to another for specialized treatment (Figure 1.2, patient *Jones*'s case). Such collaboration within or among healthcare organizations has been shown to facilitate cost-effective healthcare services.

Healthcare involves several types of MDTs with various characteristics, among which is a multi-professional team (e.g., a multidisciplinary care team in the intensive care unit [207]) consisting of physicians, nurses, and other healthcare professionals like social workers, respiratory therapists, pharmacists and administrative staff. Moreover, a geographically distributed team may consist of geographically co-located teams, for instance, multidisciplinary cancer treatment teams [132]. Regardless of their type, MDTs appear to share certain characteristics [250]. MDT members have specific roles and interact with each other to achieve a common goal [281]. The roles of healthcare professionals vary between and within teams



(a) Resource usage in isolation (b) Resource sharing environment

Figure 2.2: Resource usage in isolation and resource sharing

at different times [255]. Examples include nurses performing colonoscopies and radiographers reading plain radiographs (X-rays). Moreover, health record access and sharing are essential requirements in daily clinical care and MDT treatment [121, 133].

One key aspect of an MDT is resource (patient's EHRs) sharing [49, 121]. To collaborate, each team member must be prepared to gather and share their findings with the other team members [281]. According to Figure 2.2, each healthcare provider initially accesses his/her own resource in isolation (Figure 2.2a). However, once collaboration is established, the process of sharing begins (Figure 2.2b). Involvement in collaborative healthcare work increases outcome quality and yields greater patient and healthcare provider satisfaction, among other advantages [112, 195, 353]. A collaborative healthcare environment also enhances the proactive care of patients with long-term, severe health problems by providing round-the-clock treatment and organized responses to help fulfill primary care requests [112, 246, 353]. Evidence shows that MDTs work decreases the number of hospital (re)admissions and improves patient satisfaction [394].

Despite the numerous advantages of collaborative healthcare work, teamwork also presents challenges to healthcare providers [130, 246, 384]. These include, coordination in terms of role formation and allocation [31, 386], conflict management and resolution [90], as well as information exchange to convey and receive the knowledge and data necessary for team coordination and task completion [295]. Team role allocation should be part of the earliest team formation stages to enable the team to develop a common vision and principles of operation as well as determine the competencies and responsibilities required of every team member [31]. Conflict can be a result of miscommunication between team members regarding their needs, ideas, or goals. Conflict management involves acquiring skills related

to conflict resolution [90, 354]. Therefore, all team members need the right skills of communication, coordination, and conflict management to enable them to function as part of a team and also to enable the team to function effectively as a unit. According to a study of 27 Norwegian trauma teams, specific teamwork skills such as leadership and communication were associated with indicators of good team performance [386]. Better performing teams exhibited more effective coordination, communication and information exchange.

There are several multidisciplinary teamwork models to enhance team coordination and management, improve care quality and increase patient satisfaction. These models include *case management model* [45, 246, 394], *shared mental model* [196, 386], *integrated care pathway model* [73, 318, 370] and *key worker model* [39, 246], among other models [337, 394]. *Case management* and *key worker models* are two well-established multidisciplinary team work models [246]. In case management, a *case manager* assigns every healthcare provider a case. The case manager is also expected to coordinate the team by developing the care plan, assess the other team members' needs (e.g., providing all health information necessary for the case) and monitor as well as evaluate the care quality [198]. The *key worker model* is not very different from the case management model, except that it operates with a shared leadership[4]. The three main roles in the *key worker model* are *team leader*, *team coordinator* and *team manager* [246]. The *key worker model* is mostly applied with triage at the point of referral. Section 2.2.1 elaborates on these two models along with the respective clinical case study.

As mentioned earlier (Figure 2.2), information sharing is vital in collaboration. In order to analyze, decide and solve a certain problem collaboratively, team members must have similar knowledge of the defining situation. This study focuses on the sharing of important health information between healthcare professionals and the authorization concerns (i.e., giving official permission to access patient's EHRs) associated with information sharing [92, 333]. Balancing between shared information and security is difficult. On the one hand, collaborative systems such as EHRs are aimed at making all system resources (e.g., patient's EHRs) available to all who need them. On the other hand, access control seeks to limit access to these resources and provide them only to those with proper authorization [359]. Authorization is among the several matters discussed in literature that must be addressed with respect to collaborative healthcare environments [17, 23, 97, 121, 210, 222, 256, 303, 312, 313, 369]. Authorization mechanisms must be in place to assist MDT work, provide timely and an appropriate access to resources and protect patient privacy.

[4]Shared leadership is a leadership style that broadly distributes leadership responsibility, such that people within a team and organization lead each other [264].

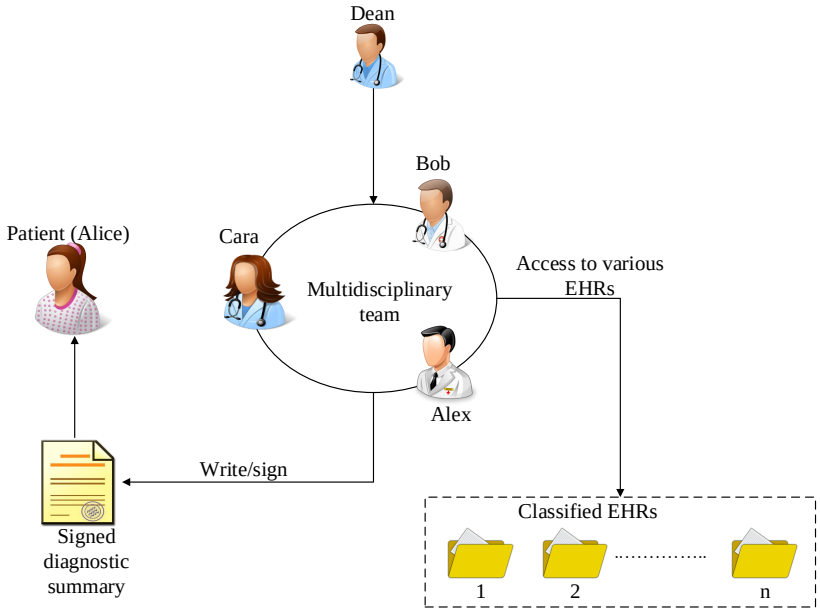


Figure 2.3: Scenario: collaboration and healthcare data sharing

2.2.1 Clinical Case Study: MDT for Cancer Treatment

To better understand collaboration in the healthcare domain, this section presents a clinical case study with examples of collaboration and healthcare data sharing.

Clinical case study 2: *Figure 2.3 illustrates a clinical case scenario adopted from [405]. A patient named Alice was recently diagnosed with gastric cancer. The only curative treatment is to surgically remove the stomach (gastrectomy). Many patients receive chemotherapy and radiation therapy after surgery to improve the chances of curing. Alice entered a cancer treatment center at a hospital of her choice (e.g., specialist hospital, Figure 1.2). Alice has a primary care doctor (Dean) whom she visits regularly. At the hospital, Alice also sees an attending doctor (Bob). Alice's health condition has caused some complications, so her attending doctor wishes to seek expert opinions and consultations regarding her treatment from different hospitals, including Alice's specific primary care doctor who has full knowledge of Alice's medical history. Note that the invited practitioners have different backgrounds, with some being specialists and others general practitioners.*

Before analyzing the security and privacy concerns in the scenario, we first briefly discuss the treatment strategy (how to build the treatment plan) and how the collaborative process is organized and managed according to MDTs work models (discussed in Section 2.2). The clinical care pathway for diagnosis and treatment maps the sequence of decisions required to identify, assess, manage and monitor the patient's case [394]. Table 2.1 is a summary of the care pathway for *Alice*'s case.

Table 2.1: Summary of care pathway model of multidisciplinary team work.

Steps	**Input level**	**Process**	**Outcome**
Step 1	1^{st} point of contact	Clinical assessment	• Identified care needs • Referral for further assessment
Step 2	Initial key worker	Identify team members who need to provide treatment	• Identified care needs • Identified MDT
Step 3		• Set goals and objectives • Interventions • Set follow-up plan	• Gastrectomy and radiation therapy or chemotherapy therapy • Identified key people responsible for each objective. Also, assigned roles for each team member • Identified follow-up plan and date

As per Table 2.1, the first contact point is usually with a primary doctor or emergency department where the primary doctor or attending physician identifies the necessary care or triages for a referral with further assessment if needed (Step 1). In case further assessment is needed, the medical coordinator (Figure 1.2) assigns the case to appropriate healthcare professionals (could be from different healthcare organization) who have the skills necessary to meet the needs and form a care team (MDT) (Step 2). Once the MDT has the full case details, the team identifies a treatment plan with objectives and goals (Step 3). The treatment plan is the care pathway process that a patient will follow on her treatment journey depending on the type and degree of problems and needs. The plan details what is going to be done, when it is going to be done and by whom. Interventions entail what is to be done to help attain the objectives [280]. There should be at least one intervention for every objective (see chapter 5 in [280] (the treatment plan)). For example, for a gastrectomy (i.e., surgical removal of part or the entire stomach), a surgeon will perform a procedure to remove a part or the entire stomach. Based on the intervention required, the *case manager* (in the case of *case management model*) or *team manager* (in the case of *key worker model*) assigns roles and tasks to each team member. As mentioned above (Section 2.2), team coordination and sharing of health information related to the case are factors that may seriously affect patient treatment.

2.2.2 Healthcare Record Sharing and Use within MDTs

Patient health information refers to information about a specific person. The usual sources for obtaining this information are the patient, family members, friends and other healthcare providers as well as the patient's health records, whether in paper

or electronic form. It has been estimated that the frequency of information needs in the course of patient care is a result of a number of questions healthcare providers ask per patient encounter [94, 100, 101, 145]. A large percentage of these information needs are not met, mostly because healthcare providers fail to find answers to the needs due to a number of barriers. For instance, the patient's answers are vague (e.g., clinical case study 1), doubting the answers given or a lack of access to resources that can directly answer the questions [110, 136, 197].

Among the expected benefits of increasing EHR use is that EHRs will become the main source of health information and vastly improve the capability of healthcare providers to find, manage and share patients' health information. However, EHRs have a remarked problem. The great amount of health information accumulating in EHRs raises security and privacy concerns regarding information access and disclosure (discussed in Section 1.1.3). Since all health information is always available, it is becoming very difficult to control access to a concrete information item required, even in relatively simple situations [283]. The health information needed by a particular member of the MDT depends on the patient's case and must be considered when identifying care needs (treatment plan), interventions and follow-up plans (Table 2.1). The information needs can be changed throughout the patient care pathway. For example, at the time of diagnosis, a member of the MDT may want little or no information about the patient's condition (family history, past surgeries, medical allergic reactions, etc.) [40, 297]. However, this may change upon adapting to a patient's case when the healthcare provider may need more information on treatment options. In general, healthcare providers seek additional information to raise their certainty about what they believe to be true and to support patients in making informed decisions [145, 329].

The most critical problem with information needs in MDT work is the lack of a common definition or categorization of what type of health information is needed during patient treatment [100, 283]. This is because, first, healthcare records contain a wide range of information, including sensitive and non-sensitive information. Second, some of this information may be needed in the care plan phase, while others may be required during the intervention phase (Table 2.1). Third, healthcare providers cannot decide what information they need and when. Healthcare providers are generally aware of the necessary information, but the requests for relevant and necessary patient health information (based on the patient's case) does arise regularly when healthcare providers see patients [94, 110, 145]. The relevancy and necessity of any information is based on the frequency of healthcare provider exposure to the problem being addressed and the type of evidence presented.

EHRs are a promising technology that makes information available instantly and accessible to healthcare providers. However, the increased availability of patient information raises ethical questions as well as security and privacy challenges concerning who should be allowed access to what information. Another question is whether particular types of information should be made available to different types of healthcare providers on a *need to know* basis, e.g. details about HIV status, history of mental health problems, genetic diseases within the family and so on [46]. These, in turn, raise technical questions about how access to EHRs can be controlled [40]. According to *Bath* [40], information needs (types of information) are not necessarily the same within a particular MDT work. In addition, as discussed in Chapter 1, standards and regulations ordain laws and policies that regulate the use and disclosure of protected health information. For example, with respect to health information, the *HIPAA Privacy Rule* permits a healthcare provider to use and disclose a patient's health information for the purpose of providing treatment based on the *minimum necessary* standard [307, 367]. Moreover, a *British Medical Association* report [65] (Access to health records: Guidance for health professionals in the UK) indicates what information should not be disclosed: (1) information that identifies a third party without that person's consent, unless that person is a health professional who has cared for the patient; (2) if in the opinion of the relevant health professional the information is likely to cause serious harm to a third party's physical or mental health; (3) information the patient provided in the past with the understanding that it would be kept confidential and (4) no information at all can be revealed if the patient requested non-disclosure.

An important conclusion in this subsection is that we are not attempting to provide an answer to what health information healthcare providers need. Rather, we endeavour to observe the importance of the availability of different health records within MDTs (see other studies reporting on the health information needs of healthcare providers [21, 64, 94, 100, 110, 136, 145, 197, 241, 297, 329, 341]). It is concluded that to provide high quality care, healthcare providers need access to patient information, including potentially sensitive patient information in many instances. However, they do not necessarily need routine access to all patient records.

2.2.3 Security Issues Arising in the Scenarios

Increasing focus on MDTs by diverse organizations leads to a greater extent of patient health information sharing and transferring within/across healthcare organizations that are utilizing an EHR system. Analyzing the case studies (clinical case studies 1 and 2) highlights that an EHR system may render all patient health

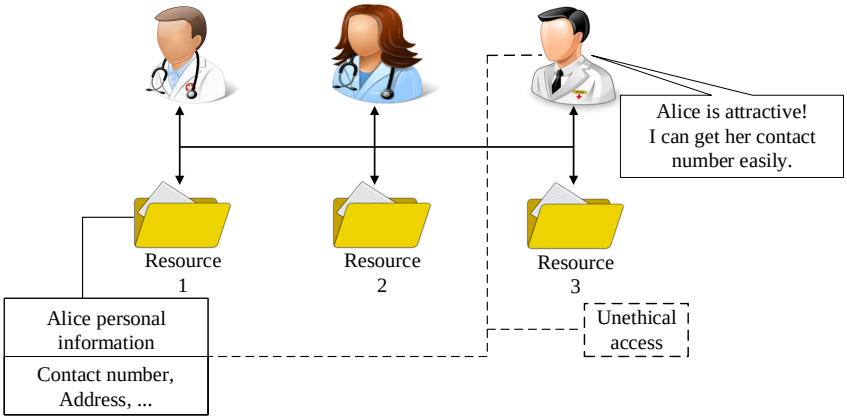


Figure 2.4: Insider threat

information vulnerable to insider threats [83, 84, 284], such as unauthorized and improper information access and disclosure [178]. Insiders represent authorized and trusted employees (current or former) in an organization who have certain privileges and access to systems [160, 178]. According to our work with the *Centre for eHealth* at the University of Agder [2], there are three entities involved in patient treatment which might pose an insider threat to patient information. The patients themselves, formal healthcare providers (e.g., doctors, nurses and other health and care support staff such as system administrators) and informal healthcare assistants, such as friends and family members who provide patient support and have very limited access to the system. This thesis only considers access by formal healthcare providers (healthcare professionals and associate professionals, Figure 1.1) working on healthcare organizations.

Unauthorized access and disclosure (intentional insider threats) can happen when someone in the collaborative team accesses shared resources for unethical reasons, for instance accessing a patient's private information for personal gain [178] (e.g., breaches at *Howard University Hospital*, *Washington* presented in Section 1.1.3). Intentional insider threats entail individuals who have legitimate access to an organization's resources and decide to abuse their privileges, thus compromising the availability, confidentiality or integrity of resources [36]. In an intentional insider threat example (Figure 2.4), it is assumed that three physicians are working collaboratively on a patient *Alice*'s case (clinical case scenarios 2) at the hospital. They want to discuss the possible treatments for *Alice*. To do so, they must analyze her health information but not her personal information. However, the 3^{rd} physician is attracted to the patient and exploits the collaborative environment to obtain her contact number without permission.

Nonetheless, one of the main causes of improper access (unintentional insider

threat) is information leakage, which can occur when a supporting party has access beyond what they actually require. Such unintentional insider threat may occur through an action or inaction without malicious intent that causes harm (e.g., patient privacy violation) [69]. For instance, in treating patient *Jones* (clinical case study 1), the main practitioner (GP) consults a specialist from another department/hospital. In doing so, improper information access might occur if the specialist (e.g., dermatologist) obtains more permissions than required.

Insider threats are of serious concern in the healthcare industry. According to a 2015 *Identity Theft Resource Centre* (ITRC) data breach report [185], 35.5% of documented breaches involved medical counterparts. Moreover, the same report in 2016 showed that the number of healthcare data breaches (377 incidents) represented 34.5% of the overall total with almost 167,263 records breached by insider theft (cf. 2016 Data Breach Insider Theft Category Summary[5]). What's notable about the insider breaches in healthcare sectors is the amount of time it took an organization to discover an incident. One danger of insider threats that occur due to collaborative effort in the EHR environment lies in their low detectability [68, 183, 311]. According to report by *Caban* (published in *Medium online publishing platform*) [209], almost 75% of healthcare data breaches go unnoticed and several insider breaches go undiscovered for more than a year (more about the details pertaining to specific breach incidents, see [209]).

Basically, an incident can happen repeatedly over an extended period of time without authorities discovering it [68, 183, 311]. This is because, understanding the intent (purpose) of healthcare providers action (e.g., accessing a certain patient's records) is a hard process. In fact, healthcare providers require legitimate access to patient EHRs to perform their jobs effectively. Therefore, actual attacks on EHRs can be attempted at any time, which makes the threat harder to detect. Considering the attack in Figure 2.4, the reason is that the 3^{rd} physician exploited his access rights from his trusted status as a healthcare professional to treat a patient. Therefore, in this case, even with forensics analysis it is hard to detect malicious actions and identify if the access was proper, with the purpose to treat, or with a malicious intent [68]. Additionally, insiders may be able to maintain good social relations with the patient and healthcare organization to utilize these in the intended exploits.

Given the severity of insider threats in the healthcare sector, a number of countermeasures have been developed and are divided into two main categories: passive and active measures [200, 262, 284]. Passive measures are more geared toward detecting the perpetrators, while active measures protect targeted assets from total

[5]http://www.idtheftcenter.org/2016databreaches.html

compromise. Access control is the most popular approach of an active form of mitigating insider threats [12, 16, 178, 309, 336, 359]. For instance, in order to secure a shared repository on epidemics, group-based discretionary access control is employed [404]. It allows certain individuals to access the data and prohibits others based on their group membership.

2.3 Access Control Requirements in a Collaborative Environment

A number of access control requirements are discussed in [11, 125, 186, 236, 253, 275, 275, 331, 359, 369]. According to the discussions in previous sections and the presented case studies, the access control requirements in a collaborative healthcare environment should include the following:

2.3.1 Security and Privacy Requirements

Requirement 1 (Personalized permission)**:** Patients must be informed of any collaboration and should have the right to choose (allow or deny) who can have access to their records [1, 109, 333]. An example is patient *Jones* (clinical case study 1) who grants the psychiatric institution permission to share his psychiatric records only with his GP (upon the GP's request). From a legal perspective (e.g., Norwegian legislation such as the *Health Personnel Act and Personal Health Data Filing System Act* [265, 387] and according to *HIPAA* in the United States [307, 367]), healthcare providers must obtain the patient's consent to be able to store or process (e.g., share and disclose) patient-related health data [109]. For instance, in Norway, the patient has a right to opt-in and opt-out of *Kjernejournal* (Section 2.1.3) [387].

Patient consent with regard to health data sharing should be given implicitly or explicitly [364, 369]. Explicit consent is when the patient is willing to make his/her own decision (e.g., when visiting a physician) to agree or disagree about who should join the treatment team and approve what information may be used or shared with the MDT. According to the EU Directive 95/46/EC [107] (superseded by GDPR [376]), the patient's agreement on access and sharing of his/her health information should be laid down in explicit consent, unless the patient is incapable of doing so. Implicit consent (also known as indirect consent) is when the patient is not willing to make their own decision and somebody else must decide on the patient's behalf (i.e., the patient allows others to make decisions on his/her behalf). The complexity of patient consent is based on the following questions. Is a given

consent once enough for all information? Should the patient give consent every time a new healthcare provider is treating the patient? What happens with patient consent when the patient changes their healthcare provider? Consider patient *Jones* (clinical case scenario 1), in case he decides to leave town and change his GP. All questions above will be answered accordingly in the Chapter 7.

Requirement 2 (Selective relevancy)**:** Certain patient information is highly sensitive. Thus, access control should facilitate withholding information that remains confidential. For example, assume patient *Alice* (clinical case study 2) has a history of a *Sexually Transmitted Disease* (STD) noted in her EHR. Questions that come to mind are: *Should this information always be available*? *Does Alice have the right to withhold this information if she thinks it is irrelevant to her current treatment?* Taking into consideration legislation to answer these questions, according to the *UK Good Medical Practice legislation* [363] (also HIPAA and *Personal Health Data Filing System Act*, etc.), healthcare providers should not access or disclose any patient health information unless relevant and necessary to the treatment. Also, the degree of patient identification in the health information must not be greater than is necessary to serve the intended purposes of the treatment.

It might be argued, *who should decide on the relevancy of the information*, *what information is relevant and when is it relevant*? As we mentioned in Section 2.2.2, not all information should be available throughout patient treatment unless legislation and patient consent permit. With regard to the principles, guidelines and recommendations [25, 44, 65, 124, 187, 259, 298] to protect patient's privacy and flow of health information, patients are the owners of their health records, and should thus have the ability to monitor and control which healthcare providers have access to their personal EHRs. While a patient might wish to share his/her record with his/her healthcare providers, he/she might not wish to allow pharmacists, billing staff or lab technicians to see any more information than is necessary [124]. Allowing healthcare providers access to all available data (when not needed) may result in losing control over information and violating patient privacy [140, 369]. In effect, the access control must be sufficiently flexible to cater to this need and support *minimum necessary* standard to use and disclose patient records.

Requirement 3 (Granularity)**:** Access control granularity refers to what is the smallest (*minimum necessary*) amount of data which can be authorized to users [301]. Access control models should be able to protect information and resources (patient's EHRs) of any type and at varying granularity levels. In MDT's work, there is a need to assign privileges to team members (healthcare providers) with accurate granularity depending on the content of the health information and work required [143]. For

example, consider patient *Jones* (clinical case study 1), a dermatologist can only access and modify patient records related to *Jones*'s case when the dermatologist is a member of *Jones*'s treatment team. Even if the system provides a high granularity of access control (e.g., for a single patient record), the assignment of each patient record to the corresponding physician would be a hard work. Therefore, a more flexible access control mechanism should allow reliable protection for shared environments and resources of various kinds as well as allow fine-grained control of access to individuals and resources [359].

Requirement 4 (Extenuating access)**:** In emergency situations, the normally delimiting nature of access control should not be in place as a barrier to medical personnel acting effectively. For example, according to Norwegian *Personal Health Data Filing System Act* [387], healthcare provider may be given access to healthcare information in the national *Kjernejournal* without the consent of the patient in emergency situations where there is serious danger to the patient's life. This means that access control should provide a means for a healthcare provider who does not have access privileges to certain information to gain access when necessary in the case of emergency or life threatening conditions [401].

2.3.2 Collaboration Requirements

Requirement 5 (Dynamicity)**:** Roles in healthcare environment often form dynamically and change constantly during MDT collaboration [255]. Therefore, access control should facilitate specifying and changing responsibilities at runtime depending on the environment or collaboration dynamics [359]. Moreover, it should permit users to take on multiple roles simultaneously and switch roles easily in different cooperation phases [331]. Access control should allow switching roles and user privileges with ease and without causing changes to policy specifications.

Requirement 6 (Flexibility, adaptability and scalability)**:** As shown earlier, a healthcare system crosses the boundary of a single healthcare institution and the number of eventual users of such system is likely to be unpredictable. On the one hand, access control should be configured to meet the needs of a large number of varying scenarios and unforeseen events of healthcare tasks and enterprise models [255]. Consequently, the access control model ought to be deployable on a large scale in order to support a large number of users and operations in a collaborative environment. Scalability, on the other hand, is an important factor in access control systems. A centralized system might not be able to follow requirements of large

scale healthcare organizations that are normally spread out geographically; thus, it is essential to consider distribution [32].

Requirement 7 (Performance)**:** The overhead of access control must be kept low to meet the real-time aspect of communication. With the high amount of data and users in a healthcare system, special care has to be taken to ensure a balance between the performance and other features of the access control scheme [255].

2.3.3 Management Requirements

Requirement 8 (Simplified user-role assignment and revocation)**:** Access control should facilitate ease of specifying and revoking user role relations (i.e., mapping roles onto a set of users). It should also support flexible administration for user-role assignment.

Requirement 9 (Fine-grained user-role assignment)**:** User-role assignment should allow different healthcare providers participating in a team to see more or less information. If, for instance, an MDT consists of three physicians working on a patient (Figure 2.4), the role permission assigned to a user should allow only the 1^{st} physician access to $resource1$ but not the 3^{rd} physician, unless relevant and necessary for the treatment.

Requirement 10 (Policy specification and maintenance)**:** Access control models are based on the specification and representation of policies that govern a collaborative environment [359]. An access control policy defines high-level rules specifying who can access a protected resource and under which conditions. Thus, the access control model should support ways of specifying policies and an appropriate syntax, pattern, or language that allows extensions or modifications in a simple and transparent manner. A high-level specification of access policy would simplify patients' understanding and healthcare providers' practices. Users should be able to specify whom they want their information (e.g., blood test) to be shared with, without entering into complex notions of access rules or security requirements [275]. Also, it is important for the access control model to provide means to ensure that the policies specified are enforced correctly [359, 395].

Requirement 11 (Usability and transparency)**:** The need of ease-of-use access control systems has been largely recognized among security researchers [104]. Usability of access control systems is even more critical and challenging to achieve in collaborative healthcare environments. The complex nature of collaborative healthcare environments, where resources can be managed by several users, makes the

specification and configuration of access preferences even more challenging. Access control systems should be unobtrusive and should not impose extra overhead on users [275]. Moreover, access control systems usually make decisions in a blackbox manner [236] and do not inform users about the privacy risks arising from access decisions. Therefore, in the context of healthcare collaboration, access control systems should be transparent to users and should allow users to understand access decisions and their effect [275].

2.4 Classical Access Control Models

Researchers in the security area have made efforts to address security challenges related to authorization and access control [104, 124, 314]. However, the proposed access control models are characterized by a considerable imbalance between security and efficiency, especially when applied in distributed environments [359, 360, 407]. Some are sufficiently efficient to fulfill the collaboration requirements but have limited security control levels [124]. In contrast, other models can meet security demands adequately but have limited authorization efficiency. Moreover, the majority of these models are related to particular applications and are implemented in centralized environments, which makes them less compatible with today's collaborative healthcare systems [222, 379].

2.4.1 Mandatory Access Control (MAC)

MAC [237, 314, 359] is characterized by a centralized access control mechanism. A single authoritative entity like an administrator manages the decision-making for granting or denying access. In effect, this central entity handles any requests for permission to an object, regardless of circumstance. To understand the implications of MAC, take a hypothetical scenario (Figure 2.5) where a user called *Sara* creates an object *X* for the purpose of work. In reality, the subject *Sara* owns the object. Common sense suggests that *Sara* should determine the permission. However, if *Jack* wishes to access the same object, he must make a request to the central entity, Admin.

Despite being strict and seemingly impractical [409], the actual rationale behind MAC is rather straightforward. The subject that owns a particular object is not necessarily the most appropriate entity to decide upon its security. This is because the owner may not fully understand the security implications of the created object. To illustrate the danger of decentralization (Figure 2.6), suppose *Sara* intends to

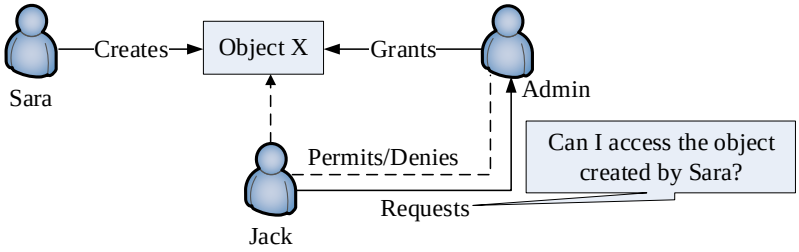


Figure 2.5: Example of mandatory access control

share object *X* with *Alex*, *Burt*, and *Cain*. Object *X* carries confidential information. Therefore, access should be provided with care. Each worker who requests access must be thoroughly examined. Being complacent, *Sara* accidentally provides access to *Cain*, who lacks proper credentials, thus potentially causing an unintentional information flow.

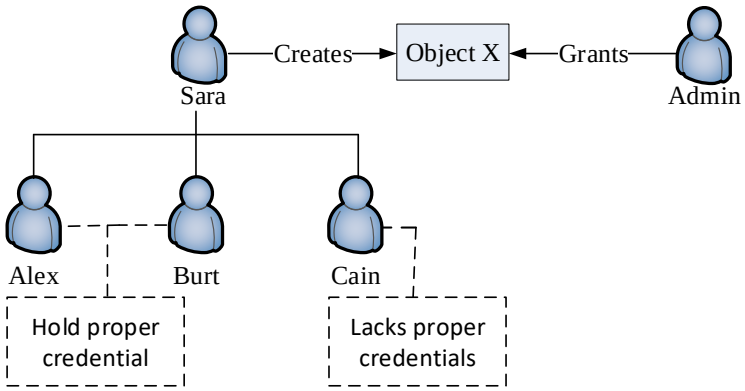


Figure 2.6: Danger of decentralization

Centralized permission granting is a simple and effective way to manage access control. Since the owner of an object may not be entirely equipped to handle the possible intricacies of security, a central entity (e.g., administrator) has the sole responsibility to do so. The central entity would then ensure that the standards are upheld consistently [49]. Consistency in permission granting is a vital aspect of security. It guarantees that permissions are coordinated coherently between subjects. This becomes more important as the number of policies rises and their interactions evolve into something more complex. Permissions conflict appear when the specifications of two or more access rules result in the conflicting decisions of permitting subjects access requests by either direct or indirect access assignments. In addition, when multiple policies are evoked for permission, conflicting decisions between policies may occur [175]. Thus, having a central decision-maker can significantly reduce the occurrence of conflicting permissions. In the case mentioned above, it is essential for the central entity to enforce the policy accordingly. As such, with

MAC, all policies can be implemented adequately over time without negligence risk. This provides the security system with more reliable fortification, thereby keeping the possibility of policy violations minimal. The caveat to implementing MAC is that it should be reserved for cases in which security is truly critical. Given that the central entity handles permission entirely, one of the main challenges is that requests processing can be time consuming as the number of subjects and access request grow.

2.4.2 Discretionary Access Control (DAC)

Unlike MAC that does not allow permission transfer, DAC [202, 314] permits the transfer of permission between subjects. This implies a somewhat decentralized access management, whereby permission to a particular object can be shared among subjects. In effect, access can incrementally spread among multiple subjects. Consider a situation when the administrator grants *Alex* permission to access object *X* as shown in Figure 2.7. If it is a form of DAC, then *Alex* receives the right to grant access to the object as well. He can now decide whether *Bob* and *Dean* are allowed to access *X*. Suppose *Alex* collaborates with *Dean*, then *Alex* can share access with *Dean* (Figure 2.7).

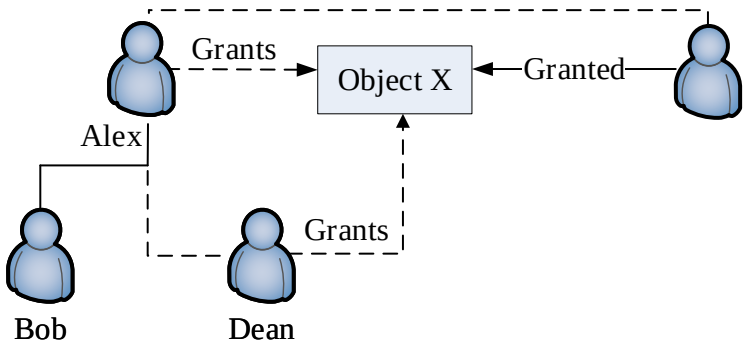


Figure 2.7: Discretionary access control

DAC simplifies the permission granting process. The administrator would only need to grant permission to a group once. Subsequently, the group can manage member access without imposing a recurring inconvenience to the administrator. In this respect, it is quite practical. By delegating the permission granting responsibility the burden can be shared. The capacity to transfer permission is rather helpful in cases where group members increase significantly in number and change dynamically as well. With DAC, any member addition or removal will not demand the administrator's intervention.

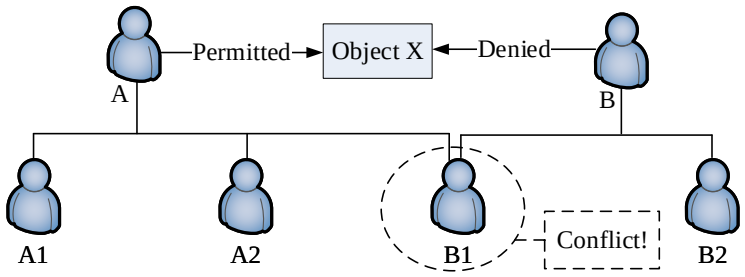


Figure 2.8: Conflict in discretionary access control

Access delegation, however, comes at a price. It complicates access coordination, which eventually results in conflicts, especially when the access granting is not managed properly [210]. Positive authorization defines what is allowed for the user, and rejects everything else. This should be contrasted with a negative authorization, which defines what is disallowed, while implicitly allowing everything else. If a user is granted both positive and negative authorizations on the same object, then we say that these two authorizations conflict with each other with respect to this user [76, 251, 308]. For instance, in Figure 2.8, supposed *A* obtained access to *X* but *B* did not. Assume further that all root node descendants inherit its ancestors' access rights. Based on the aforementioned assumption, it is quite apparent that *A1* and *A2* also obtained access to *X* but *B2* did not. A conflict would inevitably arise for *B1*, which is the descendant of both *A* and *B*. Here, the resulting access for *B1* can be unpredictable if the contradiction is not resolved adequately, posing a detrimental and hard-to-contain risk to access control.

2.4.3 Role-Based Access Control (RBAC)

In real life, permission to use a certain resource is usually granted based on the individual's role in the organization. For example, consider a hypothetical organizational chart of a hospital (Figure 2.9), where two types of clinical work are seen, namely pediatric and surgical. Thus, physicians would intuitively obtain access according to this demarcation. Physicians from the surgical department receive access to resources there as opposed to resources from the pediatric department. Granting access on the basis of the role asserted by personnel within the medical facility is the core idea behind RBAC [126, 128]. Access control strategy is encapsulated in various components of RBAC such as role-permission and user-role relationships. These components are configured by an entity (e.g., system administrators), collectively determine whether a particular user will be allowed to access a particular resource in the system [272, 317].

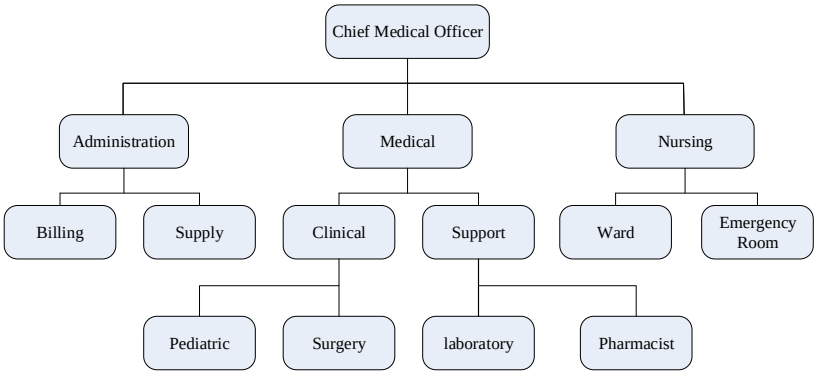


Figure 2.9: Example of hospital organizational chart

RBAC promotes the management of related permissions instead of individual ones. The sets of permissions are compiled under a particular role. Consequently, all permissions are managed based on the role itself. Any changes to the permission within the role will impact the subjects assigned the corresponding role. For instance (Figure 2.10), suppose that a collection of information on toddlers who stayed at the hospital is kept in *File-ToddlerInformation*. All permissions to access *File-ToddlerInformation* are kept under the Pediatrician role. *Susan* is assigned the Pediatrician role. Thus, she has full access to *File-ToddlerInformation*. Now if *Emma* also joins the pediatric department, she can gain access easily if she receives the same role. If a nurse named *Jenny* is delegated here, she should not have full access to the resources. A new role of Pediatric Nurse can be defined and employed for *Jenny*.

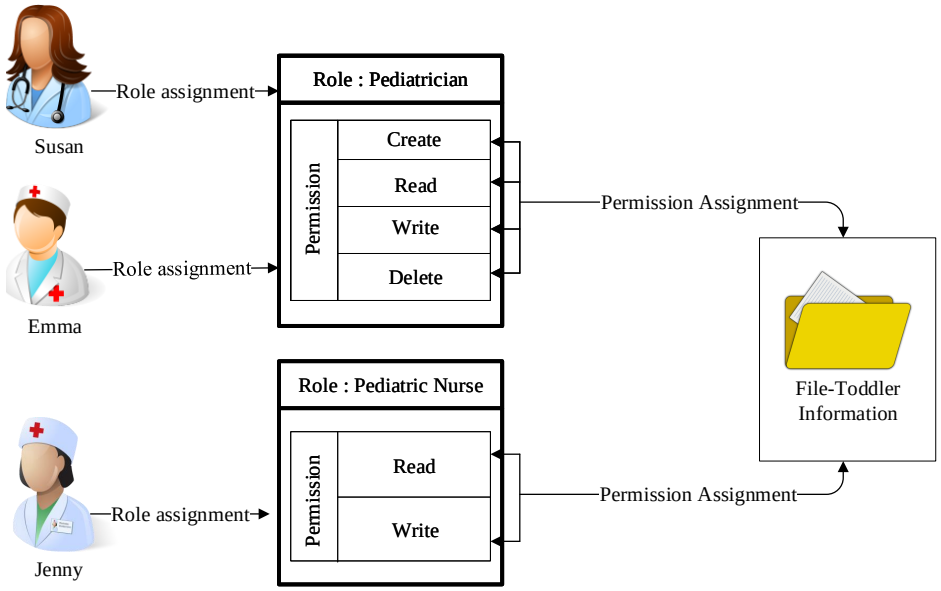
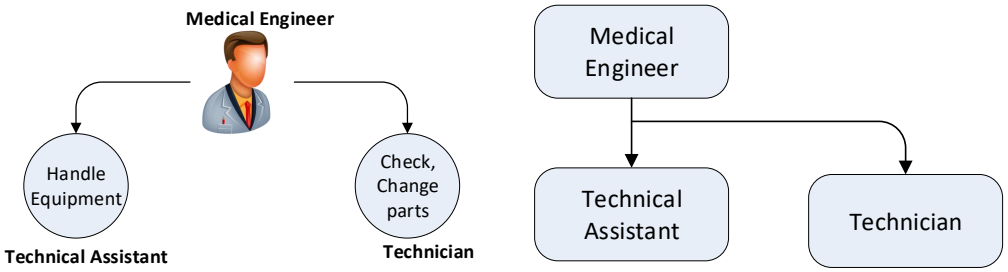


Figure 2.10: Example of role-based access control



(a) New role configuration

(b) Role hierarchy for medical engineer

Figure 2.11: Possible configuration of new roles and role hierarchy with RBAC

Role hierarchy [315, 398] is an important concept in RBAC. It states the capacity of a role to inherit the permissions encapsulated from another role. For instance, suppose there exist two roles called physician and specialist at the hospital. The specialist's role supersedes the physician's. As such, the specialist inherits the role of the physician. Inheritance is not necessarily exclusive [123]. Therefore, a role can inherit multiple roles simultaneously. The inheriting role will combine the ancestor roles' permissions. Although quite powerful, role hierarchy can induce unwanted conflict in access control [397]. To explain this predicament, consider a policy that states the need for a physician to countercheck the advice a consultant gives. The advice can be endorsed only if it is found valid. Now what could happen if the specialist inherits the roles of both physician and consultant? The specialist can give the advice, as well as check and endorse it altogether. This violates the essential purpose of the policy completely, which explains why separation of duties (SoD) [215, 217] is critical in policy making and implementation [194]. It strictly prohibits a particular role from inheriting both roles that perform and validate the processes in a policy.

Despite the possibility of conflict, RBAC is a popular access control model. This is perhaps due to the convenience it offers; it simplifies policy management and permission granting practically. Instead of having to evaluate a subject and then grant each permission individually, the task can simply be done by assigning the subjects appropriate roles. For instance, consider a new role, *medical engineer* (Figure 2.11a). The work involves equipment maintenance at the hospital. In terms of tasks, the medical engineer must be able to do two things: handle the equipment, and also check and change parts. The first task covers the technical assistant role while the second mostly overlaps with the technician role. This is commendable since the new role does not require carefully configuring a new set of permissions. Instead, it can be formed easily by inheriting the roles of technical assistant and technician (Figure 2.11b) .

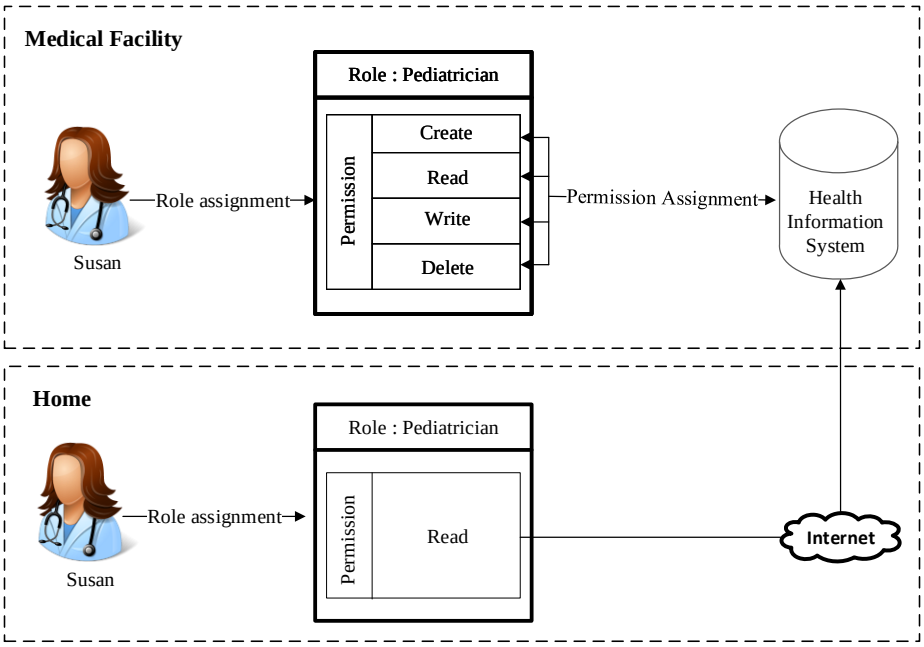


Figure 2.12: Limitation of role-based access control

Although RBAC offers expediency in terms of permission assignment to a subject, the model sorely lacks granularity [17]. In other words, it is quite difficult to define access when considering other relevant aspects beyond the one specified by the role [218]. The actual implication of this limitation requires a realistic scenario to become apparent (Figure 2.12). Suppose that *Susan* receives the pediatrician role. The permissions related to this role are secure only if *Susan* accesses the system from the medical facility itself. However, if she wishes to do so from home, permission should be limited only to the read operation. Redefining access in RBAC through other factors such as context (e.g., time and location) can be rather complicated, which is one of the motivations for developing ABAC [125, 173].

2.4.4 Attribute-Based Access Control (ABAC)

ABAC incorporates the highest degree of control and granularity with regard to defining policies for resource access. Permission can be granted based on many factors. Here, the combination of values connected to a particular attribute can serve as the conditions for authorization. More specifically (Figure 2.13), ABAC requires the establishment of work dimensions, attributes and values. The common dimensions are subject, object, action and environment. The object here refers to the specific resource that the subject accesses. Each dimension has a set of attributes and values that specifically define its meaning [218].

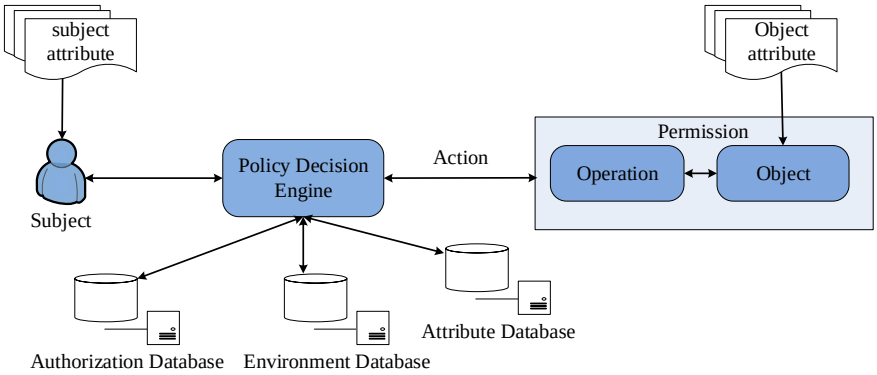


Figure 2.13: Attribute-based access control

Now suppose a new policy is needed (dilemma shown in Figure 2.12). A pediatrician can only read resources from the health information system if accessed from home. To do this, the environment can be defined at a deeper level. It is possible to use the subject's IP address to discern locality. For example, if the IP address is "$192.168.*.*$", it is considered local access; else, the resource is accessed externally. Note that an IP address is not the most secure way to determine the environment, but there are other environment attributes that can be used to allow taking the subjects' physical location into account when determining their access privileges [24, 362, 397] (more information on the application of location based RBAC in healthcare environments can be found in [156, 157]). ABAC can thus easily solve the earlier dilemma of redefining RBAC to accommodate different situations (Figure 2.14). Observe that *environment.IPAddress* serves as the deciding feature to distinguish whether access is from the medical facility or home.

Attribute-Based Access Control

```
if (subject.role== Pediatrician) and (object==healthInformationSystem) then
if (environment.IPaddress==192.168.*.*) then
        Action=grant full access        // Access from medical facility
  else
        Action=grant read access        // Access from other location
  end if
end if
```

Figure 2.14: An example of ABAC solution

The versatility of ABAC in handling diverse security requirements is a compelling reason that promotes its usage in healthcare information systems [220]. In devising more reliable protection against improper access to confidential information, ABAC can be employed to analyze the integrity of workflow conventions [380]. Analysis is aimed to discover any discrepancy between the intended

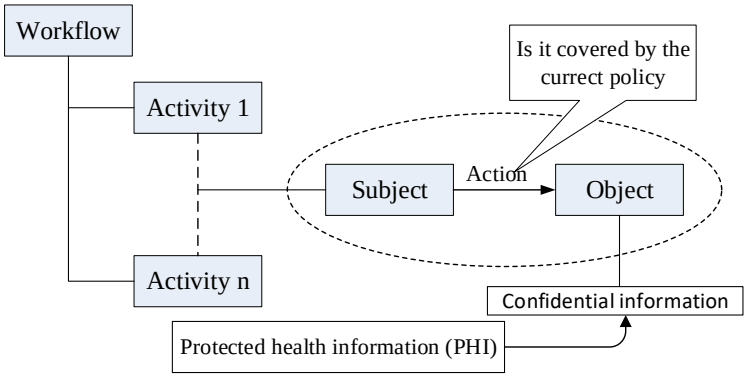


Figure 2.15: ABAC and workflow analysis

security and the one actually implemented within the healthcare facility. For analysis, each workflow is decomposed into corresponding activities (Figure 2.15). Each activity is further reduced into access control dimensions (subject, object, action and environment). The activities and resources are cross-examined exhaustively in every step of the workflow. This way, it is possible to accordingly identify any form of access that is not covered by the current policy.

Notwithstanding all the advantages of ABAC, it suffers from paralyzing complexity due to policy specifications and maintenance. The defined policies can be highly complicated as the number of dimensions and attributes further increases to cater to varying security cases. Therefore, it is imperative to test and verify the ABAC rules frequently to prevent a cascade of faulty rules from corrupting the system [220]. Granularity and manageability are documented to be inversely proportional to one another [359, 372], whereby higher granularity in security invariably implies more complex management. This is apparent in ABAC, which offers higher control or granularity at the expense of lower manageability. On the other hand, RBAC evidently provides less granularity for better manageability.

2.5 Extended Access Control Models

2.5.1 Team-Based Access Control (TMAC)

RBAC models define groups on the basis of users having the same role. Teams, on the other hand, appear to be a more natural means of grouping users in an organization and associating a collaboration context with the activity to be performed [356].

The TMAC model [356, 359] (Figure 2.16) defines two aspects of team collaboration: user context (i.e., specific users having roles in the team) and resource context (i.e., specific resources required for team collaboration). However, the major

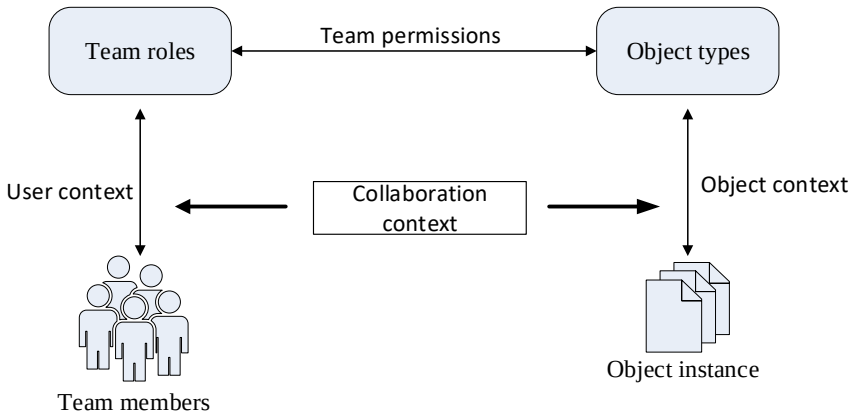


Figure 2.16: TMAC concept

flaw with this model is that all team members will obtain the same (all) team permissions [408], whereas in MDT collaboration, team members require different permissions for different resources [359, 379]. Context-based TMAC (C-TMAC) [139] is an extension of TMAC in that it uses other contextual information such as time and location. However, it inherits the drawbacks of TMAC of strictly defined team permissions.

2.5.2 Task-Based Access Control (TBAC)

The TBAC [357] model is an extension of traditional subject/object-based access control models as it includes domains containing task-based contextual information. TBAC is a dynamic access control technique, whereby access rights are not granted to subjects but rather to tasks in steps related to the tasks' progress. Each step is associated with a protection state containing a set of permissions.

Although the TBAC model tends to be flexible, it has several weaknesses when utilized in healthcare systems [359]. TBAC is limited to contexts related to activities, tasks, or workflow progress, and is implemented mainly by keeping track of permission usage and validity. Permissions are activated and deactivated in a timely manner, based on the activities or tasks. If resources (patient's EHRs) in healthcare are defined as tasks that align to business processes, the policy authors (or the resource's owner) cannot provide proper access restrictions as he/she is concerned with information that several tasks might access [369]. Moreover, complex policy specification, policy management, and authorization privilege delegation as well as revocation are very primitive.

2.5.3 Bilayer Access Control (BLAC)

Although RBAC and ABAC have their strengths, their limitations have led to a *National Institute of Standards and Technology* (NIST[6]) call [218] for the development of a policy-enhanced RBAC model, which incorporates attributes while maintaining RBAC's advantages. BLAC is a two-step method proposed to integrate RBAC with ABAC model in two multilayer to control the degree of granularity [17, 385].

An example of BLAC is proposed by *Alshehri et al.* [15, 16, 17], which enforces a two-layer access control that applies RBAC and ABAC. An access request is checked against pseudoroles, i.e., the list of subject attributes (first layer), and then against rules within the policies (second layer) associated with the requested object. A pseudorole is not a real role that is traditionally defined as a job function. Subjects' attributes are used to generate pseudoroles. They are categorized as "static" (when the attribute values do not typically change) and "dynamic" (when the attribute values change frequently). Policies make use of static and dynamic attributes to constrain pseudoroles. Despite the advantages of BLAC, it is not exclusively tailored for collaborative healthcare systems. BLAC is not meant to focus on supporting collaboration and coordination work.

Consider the case study given in Section 2.2.1 to appreciate the limitation of BLAC in managing problems potentially arising with regard to collaborative work. Suppose a physician from the primary care unit/center requires the help of another physician (e.g., gastrologist) from another department/healthcare center. Assumed in the policy prior to collaboration, only the physician in the primary care unit/center has access to reading the object or resource. Therefore, any access request by the gastroenterology department physician would be denied. This can be visualized better by studying the decision logic and process in BLAC (Figure 2.17). The access decision engine always checks the pseudorole's validity first. The physician from the gastroenterology department would have to pass the initial validation for being a physician. However, when the engine discovers that the physician's department is not primary care, access consideration halts immediately.

In order to solve this problem, BLAC recommends a modification that allows cardiologist to read data created by the primary care physician using *collaborator's subject ID*. However, enabling proper access to the object based on the collaborator's subject ID is somewhat complicated. It is difficult to define the implications of collaboration on the rule itself because it is structured by subject, object, action and environment. Therefore, a new attribute is introduced known as the *collaboratorId*.

[6] NIST is a measurement standards laboratory, and a non-regulatory agency of the United States Department of Commerce (https://www.nist.gov/)

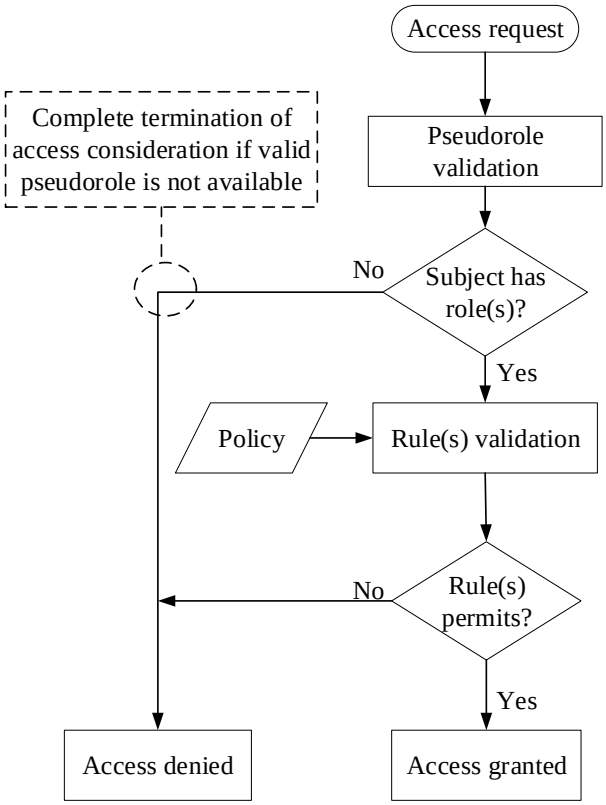


Figure 2.17: Flow of BLAC for invalid role

This new *collaboratorId* attribute should be assigned only on two conditions: the objects are created by the physician and are necessary for collaboration. However, this is a rather tedious process because it involves the additional task of security management. For convenience, suppose that all objects created by the physician in the primary care department are updated with the *collaboratorId*. Updating the objects with *collaboratorId* implies that the cardiology department physician can now read every object created by the former physician. This is true regardless of each one's purpose in the collaboration. Thus, if a confidential object is created by the primary care physician for the purpose of a crime investigation, it is visible to the collaborating cardiology physician as well.

BLAC approach has supported dynamic roles approach, which uses attributes to assign roles to subjects. However, the drawbacks of this approach include RBAC limitations such as the lack of granularity and lack of dynamic adaptability. Second, the attribute approach defines roles as another attribute of subjects, thereby inheriting the disadvantages of ABAC without any of the advantages of RBAC. Apart from difficulty controlling the scope of access, employing BLAC for collaboration can also be a source of additional complexity in constructing the rules of a policy.

2.5.4 Comparison of Access Control Models

We described several access control models proposed so far. In this section, we evaluate these models against a set of access control requirements (Section 2.3) relevant to access control models in collaborative environments.

Table 2.2 summarizes the discussion and comparative analysis of the DAC, MAC, RBAC, ABAC, TMAC, TBAC and BLAC models. The table uses comparative terminology including "Low," "Medium," and "High," descriptive terminology such as "Simple," "Complex," "Static" and "Dynamic," as well as standard terminology "Yes" and "No." The comparative terminology indicates the degree to which the requirement is supported. For example, DAC and traditional RBAC do not support selective relevancy in decision-making, whereas the MAC model supports varying degrees of selective relevancy using security labels. Moreover, the RBAC model greatly simplifies user-role assignment for administrators and it uses a static security mechanism. Yes and No are used whenever possible to indicate whether the access control model facilitates the concerned requirement. Furthermore, ABAC incorporate the fine-grained user-role assignment requirement, whereas RABC and other models do not.

2.6 Reflections on the Evolution of Access Control Models

There are numerous models for access control with emphasis on different security aspects (Table 2.3), each of which recommends a unique model to deal with the given problems. Their characteristics might overlap to a certain extent, but this does not dilute their synergistic contribution as a whole. These access control models share certain similarities, first of which is that all emphasize access management by one key concept such as roles, attributes or tasks. Secondly, each concept (roles, attributes, tasks, etc.) has its own rules that determine what permissions to grant to the shared resources.

According to our survey and others [124, 129, 181, 314], most studies rely on RBAC as the main access control model. Users (e.g., healthcare providers) who access EHRs acquire a number of roles that the system administrator predefines under certain permissions and restrictions. However, many of these access control models do not support data collaboration policies. The following subsections briefly describe and compare access control solutions.

Table 2.2: Comparison summary of different access control models

Requirements	**Access control Models**						
	MAC	DAC	RBAC	ABAC	TMAC	TBAC	BLAC
Requirement 1: Personalized permission	-	-	-	-	-	-	-
Requirement 2: Selective relevancy	High	Low	Medium	High	Low	Medium	Medium
Requirement 3: Granularity	High	Low	Low	High	Low	High	Low
Requirement 4: Extenuating access	-	-	-	-	-	-	-
Requirement 5: Dynamicity	Static	Dynamic	Static	Dynamic	Static	Dynamic	Dynamic
Requirement 6: Flexibility, adaptability and scalability	Low	Low	Low	High	-	-	Medium
Requirement 7: Performance	-	-	Medium	Complex	-	-	Complex
Requirement 8: Simplified user-role assignment and revocation	No	No	Simple	Complex	Simple	-	Simple
Requirement 9: Fine-grained user-role assignment	No	No	No	Yes	No	No	No
Requirement 10: Policy specification and maintenance	Complex	Simple	Simple	Complex	-	Complex	Complex
Requirement 11: Usability and transparency	Complex	Simple	Simple	Complex	-	-	Complex

Table 2.3: Classification of access control models

	Access Control models		
No	Emphasis	Description	Reference
1	Role	Subject's role decides the permission to operate on a resource.	[222, 304, 305, 379]
2	Attribute	The subject, object and environment attributes determine the permission.	[27, 192, 220, 309]
3	Bilayer	Integrating the role and attribute in two layers to control the degree of granularity.	[15, 17, 218]
4	Context	Enforcing access control from the context dimension, which includes the team, location, time, platform, trust and activity.	[191, 213, 255, 256]
5	Purpose	Cross checking the information usage with its purpose to ascertain proper resource access through time.	[71, 279, 377, 400]
6	Behavior	Dynamic user behavior over time as well as overall pattern to decide the access permission.	[113, 401]
7	Task	Permission is given only during task execution, for which the resource is needed.	[102, 270, 408]
8	Workflow	Access control changes in accordance with the point of execution within a workflow.	[179, 201, 310]
9	Event	Managing access control by capitalizing on events and term rewriting.	[47, 48, 52]
10	History	Analysis of previously granted and denied access to resources in deciding current permissions.	[35, 293]
11	Adaptiveness	Access control would automatically evolve when the subject interacts with the object.	[234, 242]
12	Agent	Harnessing intelligent agents to assist the progressive mechanism of access control.	[153]
13	Knowledge	Using semantics and ontologies to address the intricate decision issues in access control.	[82, 162, 252]
14	Profile	Analyzing the user's profile, such as the tendency, frequency and duration of accessing a resource.	[258]
15	Trust	Trust marks the degree to which an element such as a subject is perceived to be safe for interaction.	[406]

2.6.1 Research Trends on Health Information Access Control

In order to overcome the challenges of health information access control and to meet access control security requirements, numerous access control models have been proposed. Most of these models attempt to optimize the security requirements with respect to collaboration and management requirements. This section reviews some of the proposed access control models.

Gajanayake, et al. [134] proposed a privacy-oriented access control model for electronic health records, which consists of four modules: RBAC module, MAC module, DAC module and a purpose-based access control (PBAC) module [377]. The researchers combined the modules carefully to fulfill the requirements of each

stakeholder. *Gajanayake et al.* focused on certain requirements of EHR system end users (e.g., healthcare providers, health authorities and patients) and designed the proposed model to fulfill those requirements. However, this model does not consider collaboration requirements. A patient's confidential information is split into four main categories: identity, general health, sexual health and mental health. Moreover, the approach is inspired by the purpose-based access control proposed by *Wang et al.* [377]. The information purpose determines the access extent, whereby for instance, highly sensitive information such as sexual health would impose stricter access in contrast to general health information. This way, it is possible to protect patient privacy in multiple layers.

Alhaqbani and Fidge [11] presented an access control model by integrating three models: MAC, DAC and RBAC. Patients and healthcare providers use MAC-based security labels to express the data field's sensitivity class. The patient maintains a DAC-style access control list (ACL), nominates his/her preferred/trusted medical practitioners and sets the security clearance for each. Finally, medical authorities employ RBAC to restrict and manage access to medical records. *Motta and Furuie* [256] proposed a contextual role-based access control authorization model aimed of increasing patient privacy and patient data confidentiality. Contextual authorizations use environmental information available at access time, like the user/patient relationship, to decide whether a user is allowed to access a given resource. *Russello et al.* [310] presented a framework that provides entities access rights on the basis of the actual task to be fulfilled by the entities as part of their duties. The framework integrates two components: an authorization module based on the Ponder language, and the YAWL workflow management system. It ensures that entities can access the resources associated with a workflow task, but only while such task is active. *Hafner et al.* [152] proposed a model consisting of the specialization of the SECTET-Framework for model-driven security for complex healthcare scenarios based on Usage Control (UCON) [276]. The authors identified a number of use cases in the healthcare domain, such as dynamic access control, delegation, break glass policy, 4-eyes-principle, and usage control.

Jih et al. [191] presented a rule-based approach to context-aware access control in pervasive healthcare. The proposed context-aware rule engine is intended to run on resource-limited mobile devices. *Bhatti et al.* [51] designed a policy-based system for federated healthcare databases. They investigated use cases introduced by healthcare standards to design a context-aware policy specification language called *XML-based generalized temporal RBAC*. The proposed language is expressive enough to capture healthcare environment requirements. *Schwartmann* [320]

proposed an attributable RBAC for healthcare. The model extends RBAC with attributes to reduce the total number of role and permission objects in security administration.

Many of the works presented provide guidelines on how to achieve a trade-off between patient privacy, sensitive information confidentiality and the need for flexible access for healthcare providers. They are sufficient to protect private healthcare information and suitable for application in centralized environments. However, some of these works remain silent on collaboration requirements. Due to the distributed nature of the healthcare domain, model dynamicity, scalability and adaptability are essential features that some of the above works do not address. In Section 2.6.2, Table 2.4 shows a comparison of the access control models presented with respect to collaboration requirements. To enable better security and to achieve collaboration requirements, the collaboration mechanism is made part of the base access control model. For instance, a particularly interesting trend is the development of collaboration-oriented access control models. *Le et al.* [222] engineered such a model specifically to facilitate information access management in the context of team collaboration and workflow. *Yarmand, et al.* [401] designed a new behavior-based access control model for distributed healthcare systems. A model for customizable access control captures the user's dynamic behavior and determines access rights accordingly. It works by building a set of expected behaviors of a particular user by analyzing daily actions in terms of time, context, and combination. Whenever the user deviates from an expected action, a flag is raised for monitoring. If the user repeatedly derails from their usual activity pattern, there is a high possibility that a threat is transpiring.

RBAC modification can have a number of implications for the access control model. It can mean that sharing would only occur under certain circumstances predefined by the model and that any collaboration beyond this boundary would be deemed unfit and prohibited altogether. Shared resources are subject to the same principle, as they are confined only to certain information types within the system. As such, access is rather constricted for everyone involved. Note that research on access control models that address collaboration date back more than a decade. The next paragraphs describe a few of these access control models.

Georgiadis et al. [139] proposed the context-based TMAC (C-TMAC) built around the integration of RBAC with TMAC by incorporating context as an entity in the architecture. The ability to integrate contextual information like time and location makes models such as TMAC flexible and expressive of various access policies, thus facilitating tight and just-in-time permission activation.

Zhou and Meinel [408] presented an access control model called team task-based RBAC (TT-RBAC) that integrates the team, task, and context with RBAC. In TT-RBAC, a team encapsulates a collection of users with specific roles and a set of roles with the objective of accomplishing specific tasks. The team tasks decide the maximum permissions assigned to the team, the team roles decide the maximum permissions the team can perform, and the team members decide who can activate their roles and perform the team tasks. What task a team member can perform is decided by what roles he/she can activate in the team.

Kang et al. [201], *Ahn and Sandhu* [7], and *Oh and Park* [270] used TBAC and RBAC to define access control mechanisms for inter-organizational workflow. Roles serve as an interface between workflows and security infrastructure specific to organizations. *Li et al.* [227, 228] proposed a decentralized security administrative model called Group-based RBAC (GB-RBAC) to address the management problems of RBAC in collaborations. The GB-RBAC model supports two levels of authorization management: global or system-level management by system administrators, and local or group-level management by group administrators. In this way, several administrative tasks for different applications can spread over to many different local administrators, and a fine-grained administration model of RBAC based on local administration policies is realized.

In conclusion, current access control models in the majority of previous studies do not support collaboration environment policies, which limits access to predefined resources in centralized environments. Addressing information access in the context of collaborative healthcare teams remains a challenge and fine-grained components need to be extended for healthcare collaborative environments [222].

2.6.2 Comparing Existing Solutions

Researchers have made great efforts to propose access control models that balance security with collaboration requirements. Numerous research trends on access control models were presented in Section 2.6.1. This section (Table 2.4) compares the trends to better understand the differences between these models. A comparison is imperative and is aimed at defining the most appropriate access control model for this work. A number of studies have presented the main evaluation criteria for access control in collaborative systems [17, 125, 255, 359], and many such criteria are defined in the current access control requirements (Section 2.3). The assessment criteria are adapted from [359] as follows:

1. **Flexibility and adaptability**: The access control model should support and

Table 2.4: Comparison of access control solutions

Access Control models	Assessment Criteria					
	1	**2**	**3**	**4**	**5**	**6**
Gajanayake, et al. [134]	Medium	Yes	No	Complex	Complex	No
Alhaqbani and Fidge [11]	Medium	Yes	Low	Complex	Medium	No
Motta and Furuie [256]	High	Yes	Medium	Yes	Yes	No
Russello, et al. [310]	Medium	Low	No	Yes	Yes	Yes
Hafner et al. [152]	Low	Yes	Low	Yes	Yes	No
Jih et al. [191]	Medium	Low	No	Low	Yes	Yes
Bhatti et al. [51]	High	Yes	No	Yes	Yes	Yes
Schwartmann [320]	Medium	High	Low	Yes	Yes	No
C-TMAC [139]	High	Yes	Yes	Complex	Yes	No
TT-RBAC [408]	Medium	High	Yes	Yes	Complex	No
GB-RBAC [227, 228]	Medium	Low	Complex	Yes	Yes	No

meet the needs of a large number of varying scenarios and unforeseen events of healthcare environments.

2. **Fine-grained control**: The access control models must be able to protect information and resources of any type and at varying levels of granularity.

3. **Team of users assignment and revocation**: The access control model should support the activities of user-role assignment for individual users and managing user-teams assignment for teams of users. In addition, the model should have the capability to revoke role/team roles assigned to users.

4. **Policy specifications**: The access control model should allow access policy specification, extensions and modifications in a simple and transparent manner.

5. **Policy enforcement**: The access control model should provide and support means of ensuring correct policy enforcement or constraint specification.

6. **Design for collaborative healthcare systems**: This criterion indicates whether the access control solution was designed specifically for collaborative healthcare systems.

2.7 Chapter Summary

This chapter presented background knowledge related to this research, beginning with a brief discussion about EHRs initiatives and healthcare collaboration. As we discussed above, the use of shared EHR opens a whole range of new possibilities for flexible and fruitful collaboration among health professionals in different health

organizations. However, there are unsolved security and privacy issues remaining. We understand from our discussion and clinical case studies that, a shared health record can include the entire health record, parts of it, or consist of documentation concerning one specific ailment. Also, we understand that the insider threat undermines the potential of collaborative environments in EHR systems and sensitive patient information needs to be protected against any unauthorized or improper access.

Currently known access control models with their pros and cons pertaining to health systems were described. Confidential information at hospitals is no longer safe, as the possibility of unauthorized and improper access has become alarming. We have highlighted scenarios where RBAC alone is not enough and it is hard to express MDT work policy with roles only. An example of such a scenario is a MDT team with a goal of patient treatment and team members who are cooperating in the patient treatment. Such scenarios, give rise to other access control models such as TMAC, C-TMAC, TT-RBAC, and GB-RBAC, among others discussed in this chapter.

We conclude that more effort should be directed to enhancing current access control models in collaborative healthcare environments. This is because, on the one hand, collaborative healthcare environments are usually designed to allow information sharing among a number of users by enabling communication among participants and providing rapid access to health information when distance is involved. On the other hand, access control seeks to ensure the availability of health information, but only to those authorized.

Chapter 3

Work-based Access Control (WBAC)

This chapter presents a detailed description of the proposed access control model. Following a short description of the WBAC model, the chapter presents the main WBAC model components, collaboration work model, proposed team role, resource types and WBAC flow model. Subsequently, we discuss the eXtensible Access Control Markup Language (XACML) profile for WBAC and informally validate the model to ensure it can fulfill the main objectives.

3.1 Overview of WBAC Model

As explained in Chapter 2, there are various models for access control that emphasize different security aspects. The main access control models are RBAC and ABAC. To combine the strengths of both RBAC and ABAC and to satisfy the access control requirements of collaborative healthcare systems, we propose the work-based access control (WBAC) by introducing the team role concept and modifying the team user-role assignment model from RBAC and ABAC. WBAC enforces a three layer access control (Figure 3.1) including RBAC, a secondary RBAC and ABAC. The reason for adding the secondary RBAC layer with extra roles extracted from the MDT work requirements is to manage the complexity of collaborative engagements in the healthcare domain. This coordination layer encapsulates policies related to collaboration and MDT work so as not to overly burden the RBAC and ABAC layers. The WBAC model is defined in terms of individuals assigned to roles or teams, team members assigned to team roles, work assigned to teams and permissions associated with roles and team roles. Role and team role are applied in conjunction with handling access control in collaborative environments.

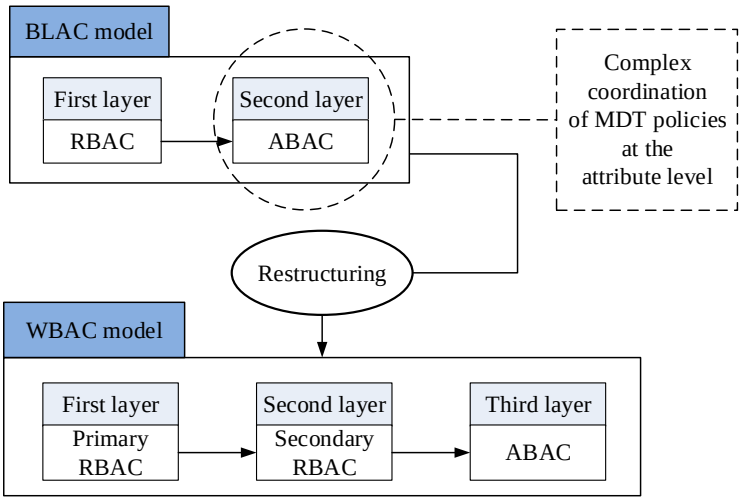


Figure 3.1: Bilayer access control and work-based access control

To evaluate an access request, the three layers evaluation procedure is preformed by the WBAC decision engine. In the first layer, the access decision engine checks the role assigned to the requesting subject to verify whether he/she has a valid role specified in the RBAC layer. If the subject hold a role with a valid permission to access the requesting object, the access decision engine checks the third layer (ABAC) to verify the rule(s) within the associated WBAC policy for additional constraints to grant or deny the access request. If the requesting subject does not hold a role or there is a rule(s) deny the request, the access decision engine evaluates the second layer (secondary RBAC) to check if the requesting subject is a part of collaborative work and hold a team role. If the subject hole a team role, the access decision engine checks the third layer (ABAC) to verify the rule(s) for additional constraints to grant or deny the access request. More details are given in the following sections.

3.1.1 Work Model for Collaboration

Work is the fundamental WBAC model entity. In this thesis, *Work* is defined as an entity comprising a collection of elements (Figure 3.2) that interact with one another in order to achieve a particular outcome successfully. In this case, *Work* refers specifically to a medical outcome (patient treatment case). As illustrated in Figure 3.2, the fundamental idea is that the *Work* demands completion and is directly linked to the *patient*, *context*, *team of personnel* and *long-term goal*.

A *long-term goal* is directly linked to an objective and an objective is broken down into a set of interventions (Table 2.1). A goal is a brief clinical statement of the condition healthcare professionals expect to change in the patient (e.g., patient recovery). On the other hand, an objective is what healthcare professionals really set

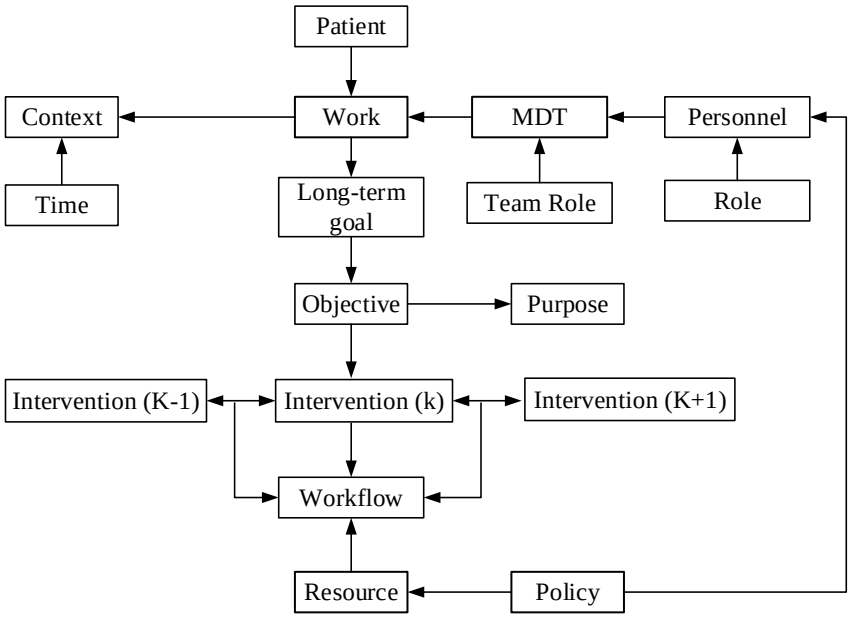


Figure 3.2: Work model for collaboration

out to accomplish in treatment and the patient ought to reach to achieve a long-term goal [280]. The difference between a long-term goal and an objective is that a long-term goal is the state that healthcare professionals intend to accomplish in general terms and an objective comprises concrete attainments that are achievable by following a certain number of steps towards a long-term goal (see chapter 5 in [280] (the treatment plan)). According to Table 2.1, objectives are concrete, whereas long-term goals are less structured. For instance, in the clinical case study 2 (Section 2.2.1), the long-term goal is to treat and cure the patient of cancer. Meanwhile, the objectives are concretely defined as gastrectomy and chemotherapy or radiation therapy after surgery to achieve the goal of *Alice*'s treatment and recovery.

Personnel entail organization employees or people engaged in an organized undertaking (Figure 1.1). Healthcare personnel are persons with special education in healthcare and who are directly related to healthcare service provision. Healthcare personnel include physicians, nurses, therapists, technicians, emergency medical staff, dental personnel, medical students, trainees and administrative staff. Administrative staff provide clerical support to healthcare professionals and executives in healthcare facilities. They are not directly involved in patient care and are not required to have an education in healthcare. As described earlier, healthcare is a team effort. *MDT* is defined as a collection of personnel in specific roles with complementary skills who are committed to a set of objectives and goals to accomplish a specific work [31, 62, 106, 195].

Each healthcare professional can be a member of a team with a special role. Some team members are doctors or technicians who help diagnose disease, while others are experts who treat disease or care for patients' physical and emotional needs. One or more teams of personnel may be involved in the *Work*. The team(s) is led by the*team manager* or *team coordinator*. Any healthcare provider joining a team interacts with other team members dynamically and interdependently towards a valued goal as well as may share a default set of permissions [31].

Role(s) and team role(s) are job functions in the context of a healthcare organization that has some associated semantics in terms of the responsibility conferred to healthcare personnel assigned roles/team roles. Role(s) is assigned to personnel and team role(s) is assigned to MDT members. To provide flexible access control during collaborative work, it is better to combine the advantages of role and team role. *Resources* in the healthcare domain can be health records in the system that personnel will access. Health record kept for each patient, maintained by the healthcare provider and it documents the patient's problems, diagnostic procedures, treatment and outcome, etc. The *policy* is high-level rules that specify how access is managed and used to verify whether any potential personnel action on resources is to be granted or denied [314, 317]. For instance, policies may pertain to resource usage within or across organizational units or may be based on *need-to-know*, competence, authority or obligation [171]. Policies can be defined on the basis of e.g., personnel attributes, geographical constraints and/or resource attributes and then formalized through a security model and is enforced by an access control mechanism [171].

3.1.2 Personnel Categories: Organizational Role

A role is a job function within an organization [388]. A role can be envisioned as a set of permissions (i.e., approval to perform an operation on one or more resources) that a subject or set of subjects can gain in the context of an organization [126, 127]. A system administrator allocates permissions to roles, including for instance a doctor's ability to partake in diagnosis, prescribe medication, and add an entry to a patient treatment record (Figure 3.3).

A role can be an organizational role, whereby the participant has a common set of permissions to perform the job function associated with the role. Hospital role examples are medical practitioners, nurses and administrators (Figure 2.9). Moreover, there can be personal roles that represent individuals and serve to create private workspaces for individuals [379]. Examples of personal roles include pediatric specialists (Figure 2.10), surgeons and pharmacists. As seen in Figure 3.3, the role of a pharmacist includes permission to dispense but not prescribe prescription drugs.

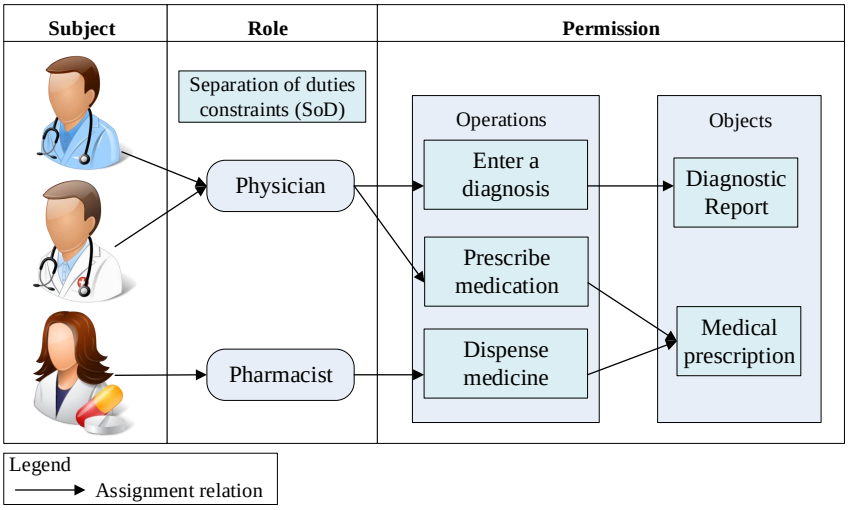


Figure 3.3: An example of subject, role and permission relationships

Users can be assigned roles statically or dynamically. The organization predefines static roles and the system administrator assigns the static roles manually to subjects based on a specific organization policy, thereby authorizing subjects to use the roles' permissions. The system administrator can also revoke membership in a static role. The main concerns regarding static roles are how to assign and revoke them from subjects and how to guarantee that subjects are assigned appropriate roles. An appealing solution is to automatically assign/revoke subjects to/from roles. Many researchers have studied the dynamic role assignment approaches. Al-*Kahtani* and *Sandhu* [10] proposed a model to dynamically assign subjects to roles based on a finite set of rules the organization defined. Moreover, *Alshehri et al.* [17] proposed a model based on the pseudorole concept (described earlier in Section 2.5.3). Although these models are intended to solve the problem of assigning subjects to appropriate roles, they still inherit the major limitations of RBAC, including lack of granularity and flexibility as well as dynamic adaptability especially in collaborative environments.

The problem of assigning subjects roles is out of the scope of this thesis. We assume that the subjects in a healthcare organization have roles, regardless of whether the roles are assigned statically or dynamically. We also assume that the WBAC model can adapt both static and dynamic user-role assignment approaches. As part of modeling and validation, we use static role assignment and assume all subjects have roles assigned. Another issue when considering collaboration among several healthcare organizations is the definitions of roles and profiles that do not exist in the organization. For example, consider Figure 1.2 (Section 1.2), if a specialized hospital wishes to grant GPs access to the hospital EHRs, it should implement GP role

and profile even though no GPs are working at the hospital. Moreover, collaborative partners should agree on the definitions and meaning of the role and profiles (e,g., healthcare professionals with a specific role has access to similar kinds of information in various systems) [369]. This situation can be more complex in practical instances where collaborative parties must agree on the definitions and meaning of roles [369], especially if a healthcare provider with a specific role wants (has) access to similar health information in various healthcare organizations. To overcome the problem of role definitions and profiling in collaboration, we propose a team role classification based on *Belbin*'s team roles [42, 247, 248] to create the correct team building process and separate permissions assigned to team members according to the required job function in the team.

3.1.3 Personnel Categories: Proposed Team Role

As we mentioned earlier that MDTs are dynamic and composed of roles determined by the patients' needs and the team members' competencies [106]. To avoid excessive information sharing (problem described in Figure 2.4), the notion of team role is used in this thesis to restrict access permissions to those individuals who not only have the right organizational roles (Section 3.1.2) but are also associated with the collaborative work via team membership.

Regarding the process of collaboration and team work, an access control model must be able to provide an efficient and secure platform for people to work together in a hospital without deterrence by restrictive access control policy enforcement [130, 222]. This can be a rather delicate situation to handle, given the fact that the fluidity of teamwork in the medical domain is often incongruent with technological security. To demonstrate this notion, consider a scenario (clinical case study 2) involving four medical practitioners who are working together on a patient's case. For the sake of securing the patient's private (sensitive) data (e.g., mental illness records) [134, 367], the collaboration must be clearly defined. By default, only the main practitioner should be aware of the patient's private information. The three other medical practitioners with supporting roles receive information based on their contributing roles. To achieve this, it is imperative to determine the finer roles of each team member. The team role of each member will subsequently determine the extent of access a member may receive.

Hospital personnel roles are often split simplistically into medical practitioners, nurses and administrators. However, their roles in a team can be further categorized using the team role theory (also called *Belbin*'s team roles) [42, 247, 248]. *Belbin*'s team role theory is useful for higher level team building processes, as it can help an

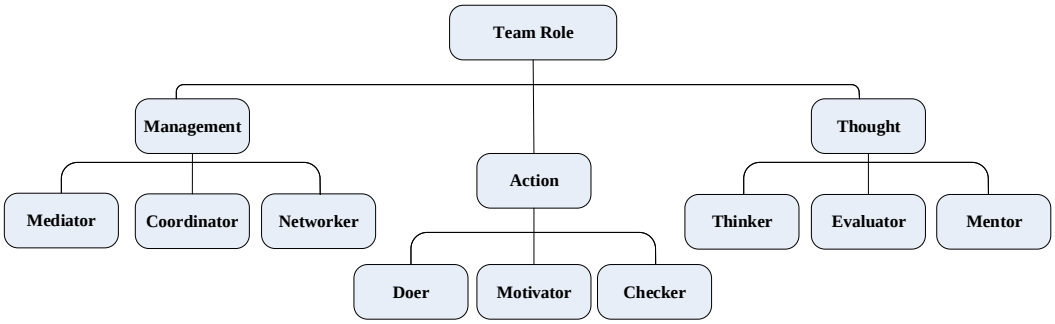


Figure 3.4: Taxonomy of team roles

experienced facilitator identify the patterns that exist within any team, thus enabling the team to manage its weaknesses and better leverage its strengths [93, 248]. The value of *Belbin*'s team role theory lies in enabling an individual or team to benefit from self-knowledge and adjust according to the demands of the situation. The team role theory contains a total of nine roles per group, which are classified into *thought*, *action* and *management*. For the purpose of this research, they are rephrased and illustrated in Figure 3.4.

- *Thought* denotes a role that is dominated mostly by thinking, analyzing problems and/or providing technical expertise [155]. To be a successful thought collaborator, the person may need to understand the medical predicament in detail without necessarily knowing the patient. A worker in this role could be involved in devising strategies to confront particular medical enigmas, make clinical decisions regarding patients' problems, formulate them into clinical diagnoses and regard interventions in terms of what to do, when to do them, and how to do them. Thus, a cardiology specialist may offer his/her expertise regarding the best practices of performing a heart transplant on a child without being involved in the actual operation.

- *Action* signifies involvement in task-related collaboration such as meeting the patient for a medical checkup. Having an action role usually implies close interaction with the patient. Nevertheless, discretion is still feasible with care. For instance, an anesthesiologist needs to only know the patient's physical characteristics and complete review of past and current medical history (e.g., a history of allergies and drug therapy) to prepare the anesthetic. Who the patient is or where the patient lives is not relevant to completing this task. This assumption is based on our review of [403] (preoperative evaluation and preparation for anesthesia and surgery).

- The *management* category comprises personnel who are mostly involved in

managing others (e.g., guiding, listening, delegating and solving conflicts). This type of collaborators are adept at coordinating teamwork that is susceptible to social or psychological challenges. For example, in conflict management, they may have to resolve series of opposing diagnoses made by medical practitioners and that may otherwise escalate into serious altercations. In this regard, such personnel's need for information is oriented inwardly. They have a greater need to know personal information about others working at the hospital rather than patient information.

3.1.4 Resource Classification

We understand from earlier discussion in Section 2.2.2 and other studies [100, 101, 197], health record classification requires a great deal of effort and skill to accomplish. This is because, first, medical records include a variety of documentation of patient history, diagnostic test results, and daily notes on patient progress and medications [355]. Second, healthcare providers cannot decide what information is really needed for patient treatment. Third, the amount of information that healthcare providers need to complete their tasks may vary greatly [21, 297]. The number of health records a healthcare provider needs to access over a certain period of time depends on many factors, including the number of patients he/she serves, the case he/she is working on, and so on. Besides, such factors vary among healthcare providers and may change from time to time. Therefore, it is very difficult to determine how much risk a healthcare provider should tolerate if they believe that knowing more information which is relevant to their patient's condition enables better decisions.

Health information classification approaches for prediction of information needs have undergone many developments [100, 101, 136, 197, 334]. Context-aware knowledge retrieval (*Infobutton* [101]), attribute selection approaches [100, 154], and machine learning algorithms [136, 334] are examples of such approaches. Despite all the strengths of such approaches, they still come with many limitations. For example, the lack of a standard to facilitate the implementation of such approaches has limited the adoption of these capabilities on a large scale [100, 101]. As far as we know, none of these solutions have been fully integrated or implemented with EHRs. For instance, according to *Del Fiol, et al.* [101], EHR developers reported fewer challenges with underlying infrastructure of *Infobutton*. The main issues were related to the nature of most EHR systems, which impose a limitations for adoption and the lack of user awareness of *Infobutton* (these issues are out of the scope of this thesis).

Furthermore, resources can be shared with other entities via several resource discovery mechanisms, for instance, the *Secure, Adaptive, Fault tolerant, and Efficient Resource Discovery* (SAFE-RD) model [328] and the *Privacy Violation Avoider* (*PriVA*) model to avoid privacy violation during resource sharing [29]. The *PriVA* model maintains a module called *TagR* (i.e., tag resource) that takes the available resource list from the resource manager and tags the resources as shareable or non-shareable [29]. For simplicity and for the purpose of this study, resources within WBAC are divided (tagged) in two different groups, mainly *protected* and *private* resources.

Resources are classified as *protected* or *private* according to the health information's relevance to a patient's case. *Protected* resources can be shared in a collaborative work. Contrary to the former type, *private* resources are highly classified pieces of health information that are shared during a collaborative work only if needed. As such, spreading access control on the basis of collaboration does not affect the *private* resources. It is meant to safeguard certain confidential information from leaking accidentally through collaborative means. In our classification, we consider the following information classification (direct information and indirect information classification is adapted from the EU Directive 95/46/EC [114] which is superseded by GDPR [55, 358, 376]. Also, in this classification, we consider the requirements of the *British medical association* report [65] discussed in Section 2.2.2):

1. Direct information that refers directly to a patient and does not involved in clinical reasoning[7], such as name, ID number and address. This type of information is always classified as *private* resources.

2. Indirect information has clinical reasoning and refers to a patient or has some relationship with a patient's privacy such as medical history. This type of information is classified as *private* resources. Psychotherapy notes are an example of such information. However, this information would be shared as *protected* resources if needed for treatment. For example, consider *Jones*'s scenario (clinical case study 1), we assumed that psychiatric notes are a type of *private* object because they are not needed for this case of treatment. But, *Jones*'s psychiatric notes would be shared if any healthcare provider thinks such information is important for the patient's case. For instance, when the dermatologist wants to know whether there is any medical information regarding allergies and medication about the patient elsewhere and the derma-

[7]Clinical reasoning is the process by which a healthcare professional interacts with a patient, collecting information, generating and testing hypotheses, and determining optimal diagnosis and treatment based on the information obtained [28, 221, 240, 290].

Collaborative Resource	Role			
Resource(1)	◎ Main	◎ Management	◎ Action	◎ Thought
	◎ Main	◎ Management	◎ Action	◎ Thought
Resource(n)	◎ Main	◎ Management	◎ Action	◎ Thought

(a) Team role simplified in tabular form

Collaborative Resource	Role			
Patient personal information	◉ Main	◎ Management	◉ Action	◎ Thought
Patient medical information	◉ Main	◎ Management	◉ Action	◉ Thought
Staff personal information	◉ Main	◉ Management	◎ Action	◎ Thought

(b) Collaborative resources and team roles

Figure 3.5: Resource classification in a collaborative environment

tologist decides to to ask patient *Jones*'s GP. In this case, we assume GPs can decide what information is needed and GP could decide if the information has serious implications for the treatment.

3. Relevant information that relates to a patient's medical case. This type of information is classified as *protected* resources. There is considerable concern in terms of "*who should decide on what objects (patient EHRs) should be shared and what should not.*" To resolve this, according to medical good practices and legislation (cf. study on the legal framework for interoperable eHealth in Europe [107]), decision regarding the necessary and relevant must be taken by policy authority and healthcare providers who possess the patient records. In our case, we assume the healthcare provider assigned as *case manager* should decide what information is necessary for the treatment. This is according to the MDT teamwork models presented in Section 2.2 where a *case manager* should assign healthcare providers a case and decide what information and resources are needed for the case. Note that, all members of the team could request any information if they think the information serves the purpose of the treatment. However, this should be done through the *team manager* or *team coordinator*.

In this thesis, a way of simplifying conflict resolution between competing policies is to utilize a tabular representation of organizing shared resources and team roles (Figure 3.5). Each resource contains four options that reflect the team roles involved. The options should not be exclusive by nature and the administrator can select none or all. Zero selection implies the resource is not open to collaborative access and can only be accessed based on user-related organizational roles (e.g., if the physician is assigned as GP for the patient) and policies. In contrast, complete selection means resources available to everyone collaborating.

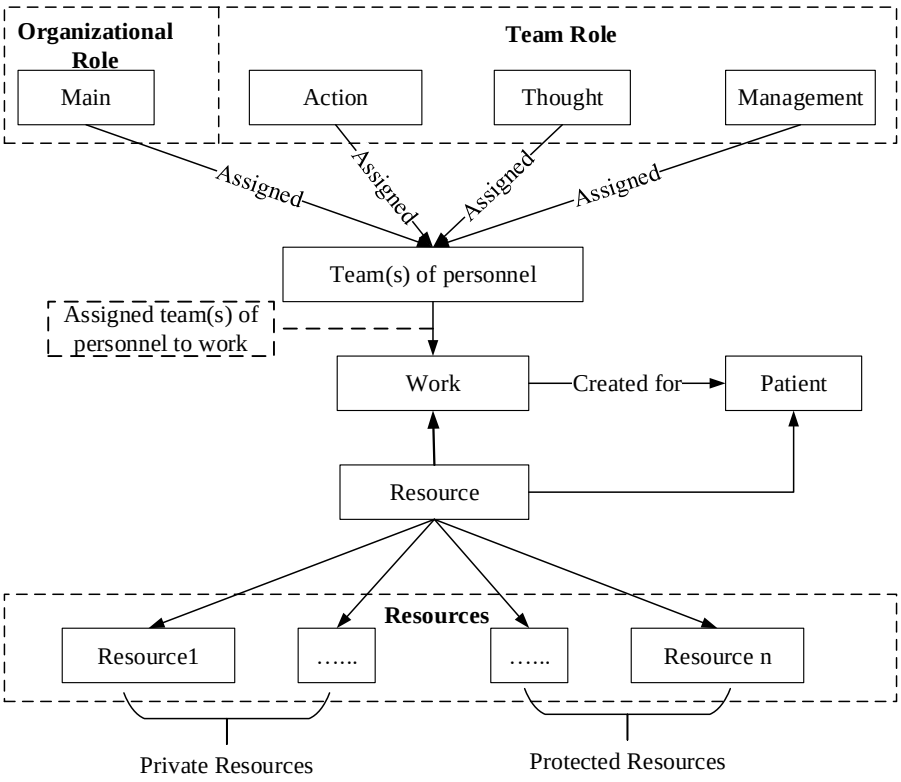


Figure 3.6: Work model for WBAC

To concretize the possibility of using tabularization in defining a collaboration policy, it is useful to consider the illustration below (Figure 3.5a), which enumerates the collaborative resources required for work in table form. Each shared resource is tied to the set team roles that can access it. In effect, the selected roles will determine the extent of collaborative access. Note that the team role for a particular resource should be set in accordance with its purpose (Figure 3.5b). A patient's personal information is vital to the main collaborator and those with an action team role. However, medical information, which might be more fundamental to treatment than personal information, should be made accessible to most team roles. It must be noted that the nature of collaboration is never free from risk. Information sharing always entails the possibility of compromise to certain security areas. As such, it is impossible to negate danger altogether.

3.2 Collaborative Work with WBAC

The work model for WBAC (Figure 3.6) postulates that the work concept can centralize the entire nature of the collaboration. It is a simplified version of the previously expounded work model (Figure 3.2) but is significantly more comprehensive

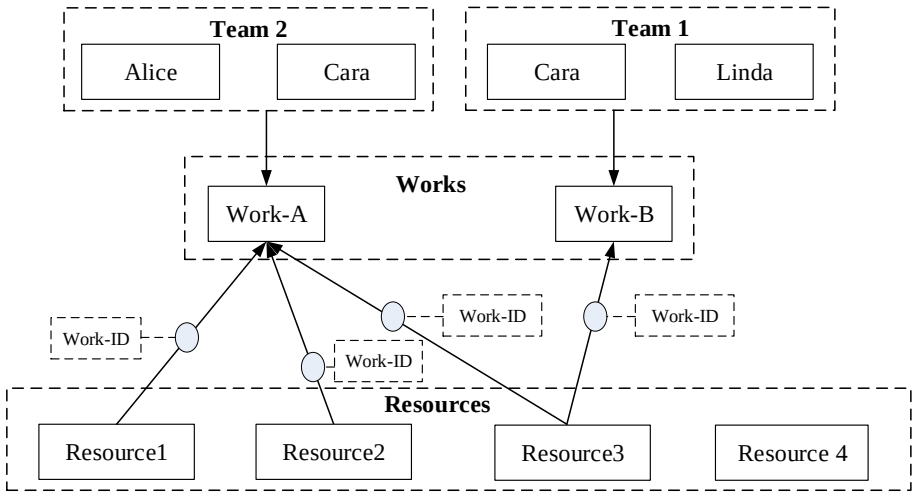


Figure 3.7: Work and shared resources

in comparison. This simplification is necessary to make WBAC sufficiently manageable from the implementation aspect. Here, each work is connected to three main components: *team(s) of personnel*, *patient* and *resource*. Hence, managing access control in collaborative work is an interplay between these components.

Every resource in WBAC is considered a collaborative entity when it is assigned a *workID*. The *workID* connects the resource to its corresponding work or project that is done cooperatively. By default, a resource does not have a *workID*, implying that it is not a collaborative resource and thus cannot be shared. To clarify the idea of managing security through a centralized work, consider the scenario illustrated in Figure 3.7. Three resources (*resource1*, *resource2* and *resource3*) are all tied to a certain work and contain a *workID* to establish this connection. However, *resource4* is not connected to any work entity. Thus, it does not contain a *workID* and can only be accessed through the main role.

As we mentioned earlier in Section 3.1, first, the subject's role (RBAC layer) is evaluated by the access decision engine. If the subject possess a role with a valid permission to process (e.g., read and write) the requested resource and the rule(s) (ABAC layer) permits the request, then the request is granted. If the requested subject does not possess a role with a valid permission or the rule(s) does not permit the request, WBAC access decision engine evaluates the assigned teams to the collaborative works to checks if the requesting subject is a member of MDT and hold a team role. If true, then the rule(s) (ABAC layer) is checked for additional constraints (e.g., if the *workID* is equal to requested resource ID). More about WBAC flow model is given in Section 3.2.3.

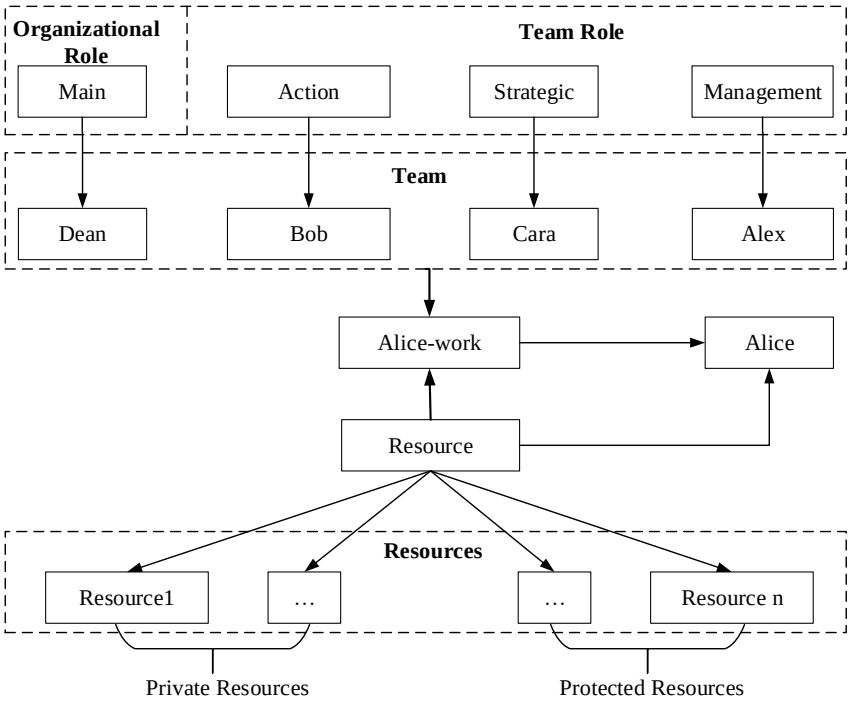


Figure 3.8: *Alice* team scenario

3.2.1 Collaborative Work Initiation

The management of a particular collaborative work requires a more rigorous explanation. First consider the clinical case study 2 (Section 2.2.1), which motivates a particular policy and its possible implications. The construction of the policy structure that fulfills the scenario in question is addressed next. To begin, the initial access control situation is that where a patient visits the hospital and registers there. The physician with whom she comes in contact as well as the nurses at the health institution gain access. As shown in the scenario, the patient's name is *Alice* and her primary care doctor is *Dean*. We assume that *Dean* has the appropriate clinical knowledge of *Alice*'s case and he has the authority to delegate the treatment or arrange a referral. Delegation involves asking a healthcare professional to provide treatment on *Dean* behalf [138]. Referral is when *Dean* arrange for another healthcare professional to provide a health service that falls outside of his professional competence [138, 289]. In either case, the act of managing the collaborative work must be defined clearly. For example, according to case management model (discussed in Section 2.2), *Dean* would be the *case manager* and decide on who should join the team and coordinate the team to develop the care plan.

According to the care pathway in Table 2.1 (Section 2.2.1), the workflow of each healthcare practitioner is as follows (Figure 3.8):

- The primary care doctor (*Dean*) cannot solve *Alice*'s case. He invites an MDT including *Bob*, *Cara* and *Alex* to help. In this team (Figure 3.8), *Dean* is the core physician in the collaborative work and serves as the team leader. He is responsible for initiating the work (treatment of *Alice*) and choosing the practitioners (group of doctors) who may be required to attend *Alice*'s consultation and treatment. This implies that he possesses the organizational role (primary physician). In other words, he owns the collaborative work that he initiated. Therefore, *Dean* gets full access in terms of patient-related information. He can access the patient's *private* and *protected* resources. Moreover, *Dean* can initially decides on what resources (*Alice*'s EHRs) should be shared with the team. keeping in mind that all other team members can request additional information about *Alice* (cf. Section 2.2.2).

- *Bob* helps *Dean* with the operational part of the case. Operation refers to a series of responsibilities that entail interaction with the patient. *Bob* needs to see *Alice* on a face-to-face basis to perform various tasks related to her recovery. In this respect, *Bob* needs to know personal and health information about *Alice* to perform his duty effectively. *Bob* is involved in the action part of the collaboration. Therefore, his team role falls under the action category.

- *Cara* has more of a strategy role. She is responsible for helping *Dean* solve the medical case. It is not necessary for *Cara* to meet *Alice* personally on a day-to-day basis. In fact, *Cara* is only required to analyze the medical situation and suggest a possible solution. *Cara*'s *thought* role in the team implies rather clearly the access she needs. Since *Cara* is predominantly preoccupied with diagnosing the disease, there is no urgent need for her to know the patient's personal information. As such, she only has access to the patient's health information as per her *thought* team role.

- With the increasing number of physicians working on *Alice*'s case, their interaction can become more complex. For instance, if *Bob* and *Cara* make competing, conflicting diagnoses, which would gain priority? This is where *Alex* comes in. He contributes to the team by coordinating the interaction of the other members by assuming the team management role (i.e., team coordinator). To work effectively, *Alex* does not really need to know the patient's personal information. However, he must be aware of the patient's health information to enable coordination. Furthermore, *Alex* must also have knowledge of work information related to the physicians. In effect, *Alex* can gain access to certain team members information and patient health information.

In addition, *Alice* may have some historical health information (mental illness or sexually transmitted diseases etc.), to which the group (or part) of specialists and practitioners do not have to have access. As per section 3.1.4, we assumed that each resource (EHRs) in the system is divided by type, mainly *private* and *protected*. The collaborative resources required for work are enumerated in table form (Figure 3.5). Each shared resource is tied to the set of collaborative roles or team roles that can access it. In effect, the selected roles will determine the extent of resources sharing and restrict the collaborative access to these resources.

3.2.2 Authorization Constraints

The authorization constraints are defined based on our team role classification and case studies as follows (more about the formal description of authorization constraints is given in chapter 4):

- The healthcare provider assigned the primary doctor role can access both *private* and *protected* resources of the patient for whom he/she is responsible.

- The healthcare provider who is a member of the care team and is assigned the *action* team role can access *private* and *protected* resources and only when necessary. In this model, we assume the healthcare provider who is assigned the *action* team role needs to access *private* resources because he/she needs to see a patient on a face-to-face basis to perform various tasks related to the patient's recovery. In this respect, the healthcare provider needs to know the patient's personal and health information to perform his/her duty effectively. Note that in other scenarios, a healthcare provider who is assigned the *action* team role may not need to know *private* patient information.

- The healthcare provider who is a member of the care team and who is assigned the *thought* team role can access *protected* resources which are approved for this particular collaborative work. This healthcare provider is predominantly preoccupied with diagnosing the disease and does not urgently need to know the patient's personal information. Moreover, the healthcare provider is responsible for helping the primary doctor solve the medical case. In fact, he/she is only required to analyze the medical situation and suggest a possible solution. In our model, personnel assigned the *thought* team role are permitted access only to *protected* resources (any resources related to the current patient case).

Table 3.1: Tabular structure of policy data for *Alice*' treatment

Subject	Role	Team Role	Object Type	Action	Permission
Dean	Primary Doctor	Main role	*Private* and *protected*	Read/write	Permit
Bob	General practitioner	*Action*	*Private* and *protected*	Read	Permit
Cara	Gastroenterologist	*Thought*	*Protected*	Read	Permit
Alex	Medical coordinator	*Management*	*Protected*	Read	Permit

- The healthcare provider who is assigned the *management* team role is responsible for coordinating the other team members' interaction by managing meetings and resolving problems with conflicting diagnoses that other team members have made. The healthcare provider does not really need to know the patient's personal information. However, he/she must be aware of the patient's health information to enable coordination. Similar to the *thought* team role, personnel assigned the management team role are permitted access only to *protected* resources. The difference between the *thought* and *management* team roles is the need for personnel assigned the *management* team role to have access to team member (healthcare provider) records. The reason is to be informed of specialized information related to the team members (physicians) in order to coordinate the collaborative work effectively.

- A collaborative work must be active for team members to be able to work on it. Assuming the value set assigned to a work is its identifier, and if there is no work, the field will not be present in a request.

According to the pathway model of multidisciplinary team work, *Bob*, *Cara* and *Alex* join the team and are assigned team roles based on the required job functions. Table 3.1 presents the structure of the policy data.

3.2.3 WBAC Flow Model

WBAC combines organizational roles (Section 3.1.2) with team roles (Section 3.1.3) to enable a multilayer role decision driven by collaboration. Process-wise, the original BLAC (Section 2.5.3) procedure is enhanced with an added decision mechanism that provides an alternative route for MDT members. Compared to the constraints that BLAC encounters in managing collaborative access control implementation, this study proposes a more dynamic policy with dual inclination in order to simplify the tasks of policy writing and policy analysis. On the one hand, the main policy specifies the actions that each user is allowed to perform on each resource

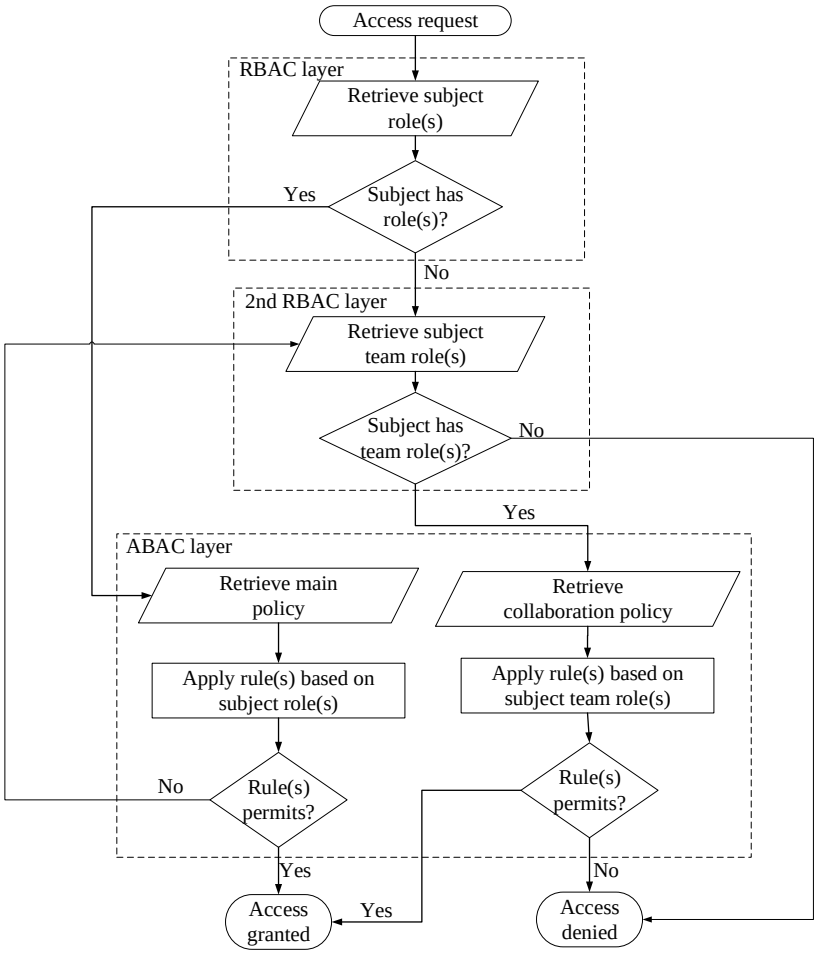


Figure 3.9: WBAC flow

based on the organizational roles. On the other hand, the collaboration policy covers any policy that mediates resource sharing based on team roles. This way, it is possible to achieve better access control management.

Similar to the core process of BLAC, the access request in WBAC first undergoes role validation (Figure 3.9). At this stage, the user's role is compared against the one defined in the RBAC layer. If the user possesses a valid role, access policy based on subject role(s) will be examine (ABAC layer) and an effect that can be either a permission or denial associated with the successful evaluation of the policy. In BLAC, failing this step results in the complete termination of decision logic (Figure 2.17). WBAC, however, treats this differently. If the request fails (user hold on valid role or there exists a rule(s) in access policy deny the access request), the resource is inspected further to determine whether it is part of collaborative work (Figure 3.9). If it is, the team role of the user in question is properly extracted and examined (2^{nd} RBAC layer). In case the user possesses a valid team role over

the resource, the collaborative policy determines the extent of access. This policy controls access granting according to the user's purpose in the team. For instance, users with the *action* team role receive more access to a patient's personal information than users with the *management* team role. This is because the *action* team role type of user has a greater need for personal information to perform their job than a user with the *management* team role.

3.3 XACML Profile for WBAC

This section presents the *eXtensible Access Control Markup Language* (XACML) profile for the WBAC model. The section first provides a glimpse of the XACML model followed by the complete syntax of XACML policy components. The second part of this section presents the WBAC policy.

3.3.1 Overview of XACML

XACML is a policy language standardized by OASIS [141, 269]. It defines the architecture, policies and messages of an access control system. XACML is a powerful and flexible policy language for heterogeneous distributed systems and is a general-purpose access control policy language [14, 125, 230]. Figure 3.10 illustrates the XACML architecture with the main entities according to references [172, 226]:

- The *Policy Enforcement Point (PEP)* is an entity that intercepts a user's request to access a resource. PEP forwards the request to PDP via *context handler* to obtain the access decision (i.e., access to the resource is permitted or denied). PEP then acts on the received decision.
- The *context handler* acts as the source of attribute values, or the data required for policy evaluation (e.g., a resource, subject, operation). It handles the canonical policy form. The requests and responses that PDP handles must be converted to the canonical form, i.e., the so-called XACML context [292].
- The *Policy Decision Point (PDP)* is used to evaluate access requests against authorization policies and to make decisions according to the information that the request contains.
- The *Policy Administration Point (PAP)* is in charge of writing and managing authorization policies in the XACML policy language and making them available to PDP.

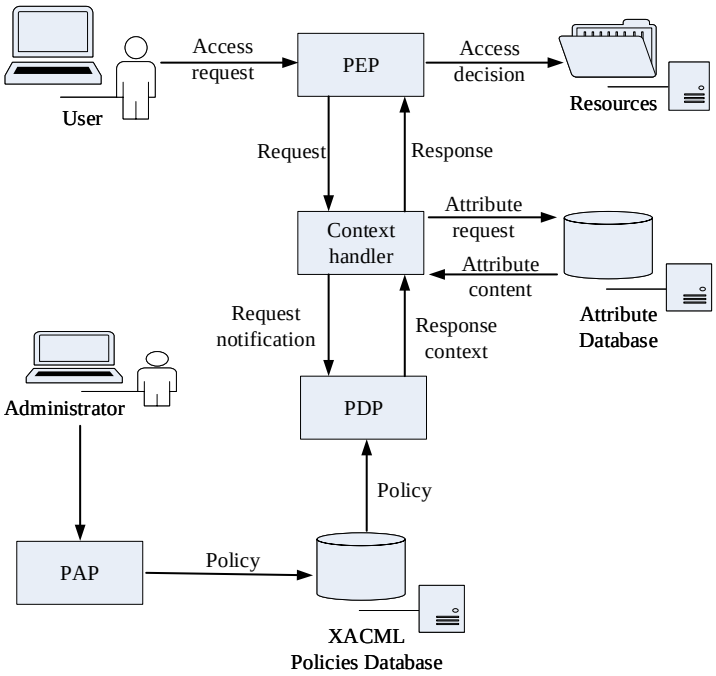


Figure 3.10: Basic XACML framework

Figure 3.10 illustrates a typical XACML scenario. The user sends an access request to perform an action on a particular resource. PEP forms a request to the context handler in its native request format, optionally including subject, resource and operation attributes. Later, the context handler constructs a request (using the XACML request language) based on the attributes of the subject, operation and resource, and sends the request to PDP. PDP retrieves the appropriate policies from the policy database and then examines the request against the applicable policies to determines whether to grant access according to the rules defined in the policies. PDP returns a decision with one of these values: *permit*, *deny*, *notApplicable*, or *indeterminate*. The answer (expressed in XACML response language) is returned to PEP via the context handler, which then allows or denies user access by translating the response context to the native response format. PDP also returns to PEP a sequence of actions called "obligation" to be performed in conjunction with enforcing the authorization decision applied to the access request.

3.3.2 XACML Components

The XACML core policy structure (Figure 3.11) consists of three components: *PolicySet*, *Policy* and *Rule*[8] [159, 226].

[8] *PolicySet*, *Policy* and *Rule* written in italic with an initial capital letter represent XACML components, whereas policy and rule written in small letters are common English terminology.

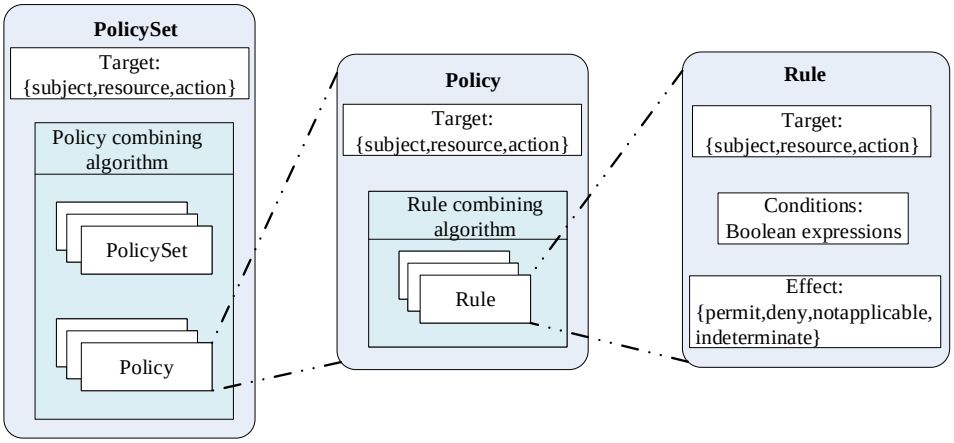


Figure 3.11: XACML policy structure

- A *PolicySet* is a collection of other policy sets or policies, characterized by a target and a policy combining algorithm
- A *Policy* consists of one or more *Rules* that apply to a certain target. Its result is computed basing on the chosen rule combining algorithm.
- The *Rule* is a fundamental component of a policy. It is composed by a target, condition and an effect. Condition is a Boolean expression that evaluates to true or false and along with the target determines the effect of the rule. Effect is the outcome of a *Rule*. The possible allowed values are usually permit or deny.

A *target* is a predicate over various categories' attributes such as the subject (e.g., user, role or/and team role), the object (e.g., allergies record), and the operation (e.g., read) of the request, specifying the type of requests to which the *PolicySet* or *Policy* must be applied. Several *Rules* are grouped and encapsulated into *Policies* and *Policies* are grouped into *PolicySets*. Correct evaluating of a rule condition returns the effect of the rule (*permit* or *deny*), while incorrect evaluation results in an error (*Indeterminate*) or the discovery that the condition does not apply to the request (*NotApplicable*). When PDP receives a request, it starts the evaluation procedure based on the retrieved *Rules*, *Policies* and *PolicySets*. Therefore, there may be conflict between multiple policies when *Policies* offer different authorization decisions. Thus, XACML provides a set of combining algorithms for combining *Rules* and *Policies* to solve a decision conflict between multiple policies. The most commonly utilized combining algorithms are as follows [159, 226]:

1. *Deny-overrides algorithm*: Combines decisions such that if any *Rule* or *Policy* evaluates denial, the decision is "*deny*."

```
namespace medical {
import Attributes.*
    * Control access to medical records
policyset medicalRecordsAccess{
    target clause resourceType== "medical records"
      apply firstApplicable
        * Doctors can view any medical records in the case of an emergency
  policy emergencyCase{
* Policy ensuring that the primary physician has clearance to access medical records
policy teamManager{
   target clause  subjectRole=="PrimaryDoctor"
   apply permitOverrides
        * A rule that permit a primary physician to read and write
       rule PrimaryDoctor{
           condition patientassignedDoctor == requestorId
           permit
       }
* Safety rule to explicitly deny access unless one of the matching rules above has been
  matched
       rule safetyHarness{
policy psychotherapyNotes{
 target clause requestorDepartment=="psychiatric" and resourceType=="psychotherapy notes"
 apply firstApplicable
   * A Psychiatry at psychiatric department can read a patient's psychotherapy notes
       rule readNote{
           target clause subjectRole=="psychiatrist" and actionId=="read"
           condition patientassignedDoctor == requestorId
           permit
       }
 * A physician can update a patient's psychotherapy note he/she wrote themselves
       rule updateNote{
           target clause subjectRole=="physician" and actionId=="update"
           condition authorID==requestorId
           permit
       }
* Safety rule to explicitly deny access unless one of the matching rules above has been
  matched
       rule safetyHarness{
 * Collaboration Policies
policyset collaborationPolicySet{
}
```

Listing 3.1: Example of a medical record access policy written in ALFA

2. *Permit-overrides algorithm*: Combines decisions such that if any *Rule* or a *Policy* evaluates permission, then the decision is "*permit*."

3. *First-applicable algorithm*: Combines decisions such that the final decision made is based on the first *Rule* or *Policy* in the policy file.

4. *Only-one-applicable algorithm*: This combining algorithm exists only to combine *Policies* in *PolicySet*, but it cannot combine *Rules*. It returns the effect of the unique policy in *PolicySet* that applies to the request, either *deny* or *permit*.

Listing 3.1 shows an example of a medical record access policy. The access policy was written in the *Abbreviated Language for Authorization* (ALFA), which is a language used to formulate XACML access control policies developed by *Axiomatics*[9]. Note, we used the feature of code folding which allows us to selectively hide and display *Rules*, *Policies* and *PolicySets*.

[9]*Axiomatics* is commercial provider of fine-grained and attribute-based authorization solutions based on the XACML standard (www.axiomatics.com).

For illustration purposes (Listing 3.1), consider an access control policy utilized in hospitals according to the *HIPAA privacy rule* [367] (i.e., only a psychiatrist designated to a patient can view the patient's psychotherapy notes). The access policy starts with *PolicySet* (line 6) to control access to medical records. The *PolicySet* uses a target (line 7) with an attribute identifier (*medical records*) to check whether the *PolicySet* is applicable to a given request. The *PolicySet* consists of *Policy* (lines 12) to permit access to any medical records in the case of an emergency. Also, *PolicySet* consists of *Policy* (lines 31-45) which ensuring that the primary physician has clearance to read and write medical records for a assigned patients. Moreover, the *PolicySet* consists of *Policy* (lines 48-70), which permits/denies any requestor access to psychotherapy notes. The *Policy* specifies a department attribute (*psychiatric*) and a resource attribute (*psychotherapy notes*) in its target (line 49). The *Policy* combines a set of *Rules* (lines 54 and 62) with the rule conditions (lines 56 and 64) to allow a psychiatrist at the psychiatric department to read/update a patient's psychotherapy notes. The *Rule* targets (lines 55 and 63) define the set of attributes to which the *Rule* is intended to apply. Besides the target and condition, every *Rule* contains the effect of access control decision (lines 57 and 65), either *permit* or *deny*. The effect propagates to the upper level policy if the *Rule*'s target matches and if the conditions are satisfied. Composite *Policy* and *Rule* evaluation is based on a particular combining algorithm that combines decisions from multiple policies and rules. In the case of the first applicable combining algorithm (policy combining algorithm, line 8 and rule combining algorithm, line 50), the combined result is the same as the result of evaluating the first *Rule* or *Policy* element in the policy list to which the target is applicable.

3.3.3 WBAC Modeling Structures

With the WBAC model, a policy is defined as a tree structure that narrows the combination of attributes presented in an access request. Access to a specific resource is granted when the whole policy tree has found possible matches (*target*) to the request. The *Rule* evaluation result is then combined upwards to the outermost policy using the combining algorithm defined at that level and is then sent back to PEP.

The XACML structure of our model is as follows:

1. Attributes are named values of known types that may include an issuer identifier or an issue date and action. Specifically, attributes are characteristics of the subject, resource, or operation. Each category usually has a set of attribute values. Examples of attributes are shown in Listing 3.2.

```
namespace Attributes {

    attribute subjectId {
        id = "urn:oasis:names:tc:xacml:1.0:subject:subject-id"
        type = string
        category = subjectCat
    }

attribute subjectRole {
        id = "urn:oasis:names:tc:xacml:1.0:subject:subject-Role"
        type = string
        category = subjectCat
    }

    attribute subjectTeamRole {
        id = "urn:oasis:names:tc:xacml:1.0:subject:subject-TeamRole"
        type = string
        category = subjectCat
    }
       attribute requestorId {
        id = "urn:oasis:names:tc:xacml:1.0:subject:user-ID"
        type = string
        category = subjectCat
    }
     attribute patientassignedDoctor {
        id = "urn:oasis:names:tc:xacml:1.0:subject:patient-assigned-Doctor"
        type = string
        category = subjectCat
    }
    attribute resourceId {
        id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
        type = string
        category = resourceCat
    }

    attribute resourceType {
        id = "urn:oasis:names:tc:xacml:1.0:resource:resource-type"
        type = string
        category = resourceCat
    }
```

Listing 3.2: Example of standard attributes written in ALFA

2. A subject (e.g., healthcare provider) is an entity that sends an access request to perform an operation (e.g., read or write) on a resource (patient EHRs). The subject is modeled based on the minimum number of attributes required to make the different decisions the policy is built to handle. Examples of subject attributes (Listing 3.2) are *subjectId*, *subjectRole*, *subjectTeamRole* and/or *patientId* (a patient for whom the physician is responsible). These are standard attributes that *Axiomatics* uses. Information about the subject also includes the team attributes for the current collaboration work (e.g., *workTeamId* and *patientWorkId*).

3. Similar to the subject, resources are elements defined by identifier/value pairs. A resource is modeled based on a number of attributes required to make different access decisions. In our model, we also consider several resource attributes (e.g., *resourceId* and *resourceType*) as shown in Listing 3.2.

4. An operation represents the operation that a subject can perform on a resource. Examples of operation include the *read* and *write* operations.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2  <!--This file was generated by the ALFA Plugin for Eclipse from Axiomatics AB
  (http://www.axiomatics.com).
3  Any modification to this file will be lost upon recompilation of the source ALFA file-->
4 <xacml3:PolicySet xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
5     PolicySetId="medical.medicalRecordsAccess"
6     PolicyCombiningAlgId="policy-combining-algorithm:first-applicable"
7     Version="1.0">
8     <xacml3:Description>Control access to medical records</xacml3:Description>
9     <xacml3:PolicySetDefaults>
10
   <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>
11    </xacml3:PolicySetDefaults>
12    <xacml3:Target>
50    <xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
51        PolicyId="medical.medicalRecordsAccess.emergencyCase"
52        RuleCombiningAlgId="rule-combining-algorithm:permit-overrides"
53        Version="1.0">
54        <xacml3:Description>Doctors can view any medical records in the case of an
   emergency</xacml3:Description>
55        <xacml3:PolicyDefaults>
58        <xacml3:Target />
59        <xacml3:Rule
113   </xacml3:Policy>
114   <xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
115       PolicyId="medical.medicalRecordsAccess.teamManager"
116       RuleCombiningAlgId="rule-combining-algorithm:permit-overrides"
117       Version="1.0">
118       <xacml3:Description>Policy ensuring that the primary physician has clearance to
   access medical records</xacml3:Description>
119       <xacml3:PolicyDefaults>
122       <xacml3:Target>
138       <xacml3:Rule
162       <xacml3:Rule
168   </xacml3:Policy>
285</xacml3:PolicySet>
```

Listing 3.3: XACML example of top-level policy set

3.3.4 Policy Set and Policy Model

ALFA (shown in Listing 3.1) supports all the data types that are defined in the XACML core specification. The native attribute values mapped directly from ALFA to XACML as shown in Listing 3.3. The WBAC policy model begins with a top-level *PolicySet* containing one *Policy* for viewing medical records in case of emergency and one *Policy* for handling a case where the subject is the patient's primary physician as well as a *PolicySet* for collaboration policies. For example, a top-level *PolicySet* (line 1 to 156, Listing 3.1) is converted to XACML containing the *Policy* for the case of emergency (lines 50 to 113, Listing 3.3) and the *Policy* for the case of primary physician (lines 114 to 168, Listing 3.3). The syntax used in Listing 3.3 and other Listings is somewhat abbreviated due to space limitations and readability. Also, we used the feature of code folding which allows us to selectively hide and display *Rules*, *Policies* and *PolicySets*.

The top-level *PolicySet* contains another *PolicySet* (line 76, Listing 3.1) for the different collaboration cases as shown in Listing 3.4. Each *Policy* in this *PolicySet* is for one specific team role and the rules that apply to this team role. A *Policy*

```
1 <?xml version="1.0" encoding="UTF-8"?>
2  <!--This file was generated by the ALFA Plugin for Eclipse from Axiomatics AB
  (http://www.axiomatics.com).
3  Any modification to this file will be lost upon recompilation of the source ALFA file-->
4 <xacml3:PolicySet xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
5     PolicySetId="medical.collaborationPolicySet"
6     PolicyCombiningAlgId="policy-combining-algorithm:first-applicable"
7     Version="1.0">
8     <xacml3:Description>Collaboration Policies</xacml3:Description>
9     <xacml3:PolicySetDefaults>
12    <xacml3:Target>
50    <xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
51        PolicyId="medical.collaborationPolicySet.thoughtTeamRolePolicy"
52        RuleCombiningAlgId="rule-combining-algorithm:permit-overrides"
53        Version="1.0">
54        <xacml3:Description>Policy for thought team roles</xacml3:Description>
55        <xacml3:PolicyDefaults>
58        <xacml3:Target>
74        <xacml3:Rule
122       <xacml3:Rule
128   </xacml3:Policy>
129   <xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
130       PolicyId="medical.collaborationPolicySet.actionTeamRolePolicy"
131       RuleCombiningAlgId="rule-combining-algorithm:permit-overrides"
132       Version="1.0">
133       <xacml3:Description>Policy for action team roles</xacml3:Description>
134       <xacml3:PolicyDefaults>
137       <xacml3:Target>
153       <xacml3:Rule
213       <xacml3:Rule
219   </xacml3:Policy>
220   <xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
221       PolicyId="medical.collaborationPolicySet.managementTeamRolePolicy"
222       RuleCombiningAlgId="rule-combining-algorithm:permit-overrides"
223       Version="1.0">
224       <xacml3:Description>Policy for management team roles</xacml3:Description>
225       <xacml3:PolicyDefaults>
228       <xacml3:Target>
244       <xacml3:Rule
293       <xacml3:Rule
341       <xacml3:Rule
347   </xacml3:Policy>
348 </xacml3:PolicySet>
```

Listing 3.4: XACML example of collaboration *PolicySet* in a top-level policy set

for *thought* team roles (lines 50 to 128, Listing 3.4), a *Policy* for *action* team roles (lines 129 to 219, Listing 3.4) and a *Policy* for *management* team roles (lines 220 to 347, Listing 3.4).

Listing 3.5 represents an example of a *Rule* combined with the *Policy* (line 114) shown in Listing 3.3. This example of rule is mapped directly from ALFA example shown in Listing 3.1 (line 38-41, Listing 3.1). This rule ensures that the primary doctor has clearance to access medical records (lines 138 to 161, Listing 3.5). The *Policy* specifies a target to the subject, according to which the policy applies to requests that a healthcare provider has issued with the purpose of accessing a resource. The value of the the policy is determined from the evaluation of the policy's rules according to the specified combining algorithm (line 116, Listing 3.5). The enclosed *Rule* does not specify any target, thus the *Rule* inherits the target of the enclosing *Policy*. This means that for every request, if the target is "*subjectRole=primary doctor*", the outcome of the policy will always be "permit" (i.e., the *Rule*'s effect).

```
114     <xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
115         PolicyId="medical.medicalRecordsAccess.teamManager"
116         RuleCombiningAlgId="rule-combining-algorithm:permit-overrides"
117         Version="1.0">
118         <xacml3:Description>Policy ensuring that the primary physician has clearance to
    access medical records</xacml3:Description>
119         <xacml3:PolicyDefaults>
122         <xacml3:Target>
138         <xacml3:Rule
139                 Effect="Permit"
140             RuleId="medicalRecordsAccess.teamManager.PrimaryDoctor">
141             <xacml3:Description>A rule that permit a primary physician to read and write
142             a patient's medical records</xacml3:Description>
143             <xacml3:Target />
144             <xacml3:Condition>
145                 <xacml3:Apply FunctionId="function:any-of-any">
146                     <xacml3:Function FunctionId="function:string-equal"/>
147                     <xacml3:AttributeDesignator
148                         AttributeId="subject:patient-assigned-Doctor"
149                         DataType="string"
150                         Category="subject-category:access-subject"
151                         MustBePresent="false"
152                     />
153                     <xacml3:AttributeDesignator
154                         AttributeId="subject:user-ID"
155                         DataType="string"
156                         Category="subject-category:access-subject"
157                         MustBePresent="false"
158                     />
159                 </xacml3:Apply>
160             </xacml3:Condition>
161         </xacml3:Rule>
162         <xacml3:Rule
168     </xacml3:Policy>
```

Listing 3.5: XACML example of *Rule* combined with the *Policy* in a top-level policy set

Similar to the primary physician policy, every *Policy* in the collaboration *PolicySet* is specified with a target to check the policy's applicability for an access request. If the policy is applicable for the request, the rules enclosed in the policy are evaluated. Finally, every *Policy* has a safety rule to explicitly deny access unless one of the non-matching rules has been matched.

3.3.5 Request Model

The XACML request contains the attributes related to the subject, resource and operation with their corresponding values. For example, in our case and as depicted in Listing 3.6, we have attribute *subjectId* and its value *Dean*, and attribute *resource-Type* and its value *AlicePrivate* as well as an operation (*actionId*) value *read*. This information is necessary for authorization decision-making.

On receiving the XACML request, the PDP starts to evaluate the request against policies in its repository and prunes the irrelevant ones by comparing the attribute values present in the request to those specified in the policies to make a decision. XACML provides two means for policies to resolve attribute values from the request; *AttributeDesignator* and *AttributeSelector*. *AttributeDesignator* allows the

```
1 <Request xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
2 ReturnPolicyIdList="false">
3 <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
4 <Attribute IncludeInResult="false"
5 AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id">
6 <AttributeValue
7 DataType="http://www.w3.org/2001/XMLSchema#string">Dean</AttributeValue>
8 </Attribute>
9 <Attribute IncludeInResult="false"
10 AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-Role">
11 <AttributeValue
12 DataType="http://www.w3.org/2001/XMLSchema#string">primary doctor</AttributeValue>
13 </Attribute>
14 </Attributes>
15 <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
16 <Attribute IncludeInResult="false"
17 AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-type">
18 <AttributeValue
19 DataType="http://www.w3.org/2001/XMLSchema#string">AlicePrivate</AttributeValue>
20 </Attribute>
21 </Attributes>
22 <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
23 <Attribute IncludeInResult="false"
24 AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id">
25 <AttributeValue
26 DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
27 </Attribute>
28 </Attributes>
29 </Request>
```

Listing 3.6: Example XACML access request

policy to specify an attribute with given identifier, category and data type. *Attribute-Selector* on the other hand provides the means to lookup the values of attributes by specifying the data type and expression [269].

3.3.6 WBAC Informal Semantics

This section presents the WBAC policy evaluation. When PDP receives a request, it starts the evaluation procedure based on the retrieved policies. PDP then returns the evaluation result as a response. Approaches for policy indexing and target evaluation are outside the scope of this work. These approaches have been discussed in the XACML v3.0 OASIS standard [269].

First, the *PolicySet* target is evaluated to determine if the policy set applies to the access request (Listing 3.6). If the target matches, the enclosed *Policies* are evaluated and the results are combined according to the combining algorithm specified in the *PolicySet* (line 6, Listing 3.3). In case the *PolicySet* target does not match the request, the returned value is not applicable; if the target is evaluated as indeterminate, then a *Indeterminate* value is returned as determined by evaluating the enclosed *Policies*.

In the original syntax of XACML, the evaluation of subjects, objects and operations are defined according to the match tables (Table 3.2). The evaluation of a

Table 3.2: Target operators

(a) *AllOf* evaluation

All Value Match			
AllOf	**Match**	**No-match**	**Indeterminate**
Match	Match	No match	Indeterminate
No-match	No match	No match	No match
Indeterminate	Indeterminate	No match	Indeterminate

(b) *AnyOf* evaluation

Any Value Match			
AnyOf	**Match**	**No-match**	**Indeterminate**
Match	Match	Match	Match
No-match	Match	No match	Indeterminate
Indeterminate	Match	Indeterminate	Indeterminate

target, described in Table 3.2, is determined by combining the results of the evaluation of its match elements. Specifically, a target matches if *all* categories it encloses (i.e., subjects, resources, operation) match. Instead, if at least one category evaluates to indeterminate then, the target is indeterminate; otherwise, the target does not match. An *AllOf* element (Table 3.2a), if all elements specified in the target match the values in the request context, the target value shall be match. Instead, if at least one of elements evaluates to no-match, the *AllOf* target is no-match. An *AnyOf* element matches (Table 3.2b) if at least one of its elements matches. If no element matches and at least one of them is indeterminate then also the *AnyOf* is indeterminate. XACML supports a wide range of (standard) matching functions (see Appendix A.3 in [269] and [230, 261] for more details on target matching). XACML use indeterminate states to handle different error types in target and condition expression evaluation, including fails in attribute retrieval due to either failure network connections or systematic errors and missing attributes in the requests. Also, in XACML, an empty target matches any request.

The evaluation of a single *Policy* is similar to that of a policy set. First the policy's target is evaluated. Then if the target matches the request or an indeterminate result is given, the policy value is determined from the evaluation of the enclosed *Rules*. The combining algorithms available in case of simple policies are the same as those for policy sets described before.

Evaluating a rule entails first evaluating its target, and subsequently, if necessary, its condition. If the target does not match or evaluates as indeterminate, it is not necessary to evaluate the condition; the evaluation yields *NotApplicable*. If the target matches, the condition is evaluated, and if satisfied, the evaluation result is the

effect specified in the rule, which is then propagated to the upper-level *Policy*. If an error occurs during rule evaluation, for instance if the target match and condition is indeterminate due to missing attributes, then the evaluation returns an indeterminate value. Notably, in our policy, if the target or condition returns *indeterminate*, then the rule evaluation returns *deny*. In our experiment, PDP is configured deny-based, which means any response that is *Indeterminate* or *NotApplicable* is seen as a deny response.

When PDP completes the decision process, the final decision that can include obligations is sent to PEP for the enforcement process. PEP does not refer to the request for its process but only to the decision statement received from PDP. In our case, we consider two possible decisions: access is either granted or denied. In fact, XACML supports extensibility to accommodate other decision results (e.g., *Indeterminate* or *NotApplicable*), which are considered denied in our case. However, *Indeterminate* or *NotApplicable* can be useful for policy testing or when they imply other actions, e.g., request reformulation and resubmission. Such aspects are out of the scope of this work.

3.3.7 Experiment and Results

The WBAC model described in section 3.3 was implemented using XACML 3.0. *Axiomatics* software development kit (SDK) with deployable *Axiomatics*-embedded PDP is used to test the policy against different requests. *Axiomatics* SDK provides complete support for all mandatory XACML features and a number of optional features. It also provides support for parsing both policy and request/response documents, determining policy applicability, and evaluating requests against policies. All standard attribute types, functions, and combining algorithms are supported.

In our experiment, the embedded PDP module is configured to be deny-based, meaning that any response which is *Indeterminate* or *NotApplicable* is seen as a deny response. We tested the WBAC policy using the attributes based on the data models shown in Listing 3.2 to build an access control policy. The PEP intercepts the requests for resource access, produces a XACML request (shown in Listing 3.6) and sends it to PDP for the actual decision-making. Upon receiving the XACML request, PDP looks up the XACML policies deployed on it and determines the ones pertinent to the specific request. It may, if necessary, query for additional attributes needed to evaluate the policies. Armed with the attributes contained in the XACML request, the attributes obtained from the database, PDP decides whether the XACML request will be either permitted or denied.

Table 3.3: Example results of requests evaluation

Clinical case scenario 2		**Clinical case scenarios 1**	
Request	**Respond**	**Request**	**Respond**
Request (*Dean*, *Private*, Read)	Permit	Request (*Saul*, *Protected*, Read)	Permit
Request (*Dean*, *Protected*, Write)	Permit	Request (*Mika*, *Private*, Write)	Deny
Request (*Bob*, *Private*, Write)	Deny	Request (*Mika*, *Protected*, Write)	Permit
Request (*Cara*, *Protected*, Read)	Permit	Request (*Carrie*, *Protected*, Read)	Permit

```
1 <Response xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
2  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3  <Result>
4  <Decision>Permit</Decision>
5  <Status>
6  <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
7  </Status>
8  <Obligations>
39  </Result>
40 </Response>
```

Listing 3.7: Decision for request in Listing 3.6 with respect to rule in Listing 3.5

Both valid and invalid values were set for the different attributes to verify that access was permitted and denied correctly. Table 3.3 provides an examples of results obtained from evaluating various test requests according to our case studies. Table 3.1 presents the policy data used as XACML input for the clinical case study 2. The evaluation of the *Dean* request in Listing 3.6 with a respect to the policy set in Listing 3.3 and rule in Listing 3.5 produces the positive response permit as shown in Listing 3.7. This means that the PDP has been capable of retrieving and evaluating all attributes specified. The PDP response will be then enforced by the PEP, whose final decision will depend on its capability to discharge the obligations.

Considering patient *Jones*'s case (clinical case study 2), we assume three healthcare providers (*Soul*, *Mika*, and *Carrie*) are working on the patient case. As shown in Figure 3.12, *Soul* is the primary doctor, *Mika* is a dermatologist who is assigned to *action* team role and *Carrie* is the medical coordinator who is assigned to *management* team role. Table 3.4 presents the structure of the policy data. The request in Listing 3.8 is made by doctor *Mika* who is a member of patient *Jones*'s treatment team. The evaluation of the request with respect to the our collaboration policy set (Listing 3.4) produces the decision reported in Listing 3.9. The resulting decision *deny* is indeed obtained because our policy allows *action* team role to read a *private* resource, but not to write.

The experiments indicated that the WBAC model granted access correctly to subjects matching the same work as the patient's resource for the expected cases. Invalid requests, such as a subject with a work team ID of 223 while the patient

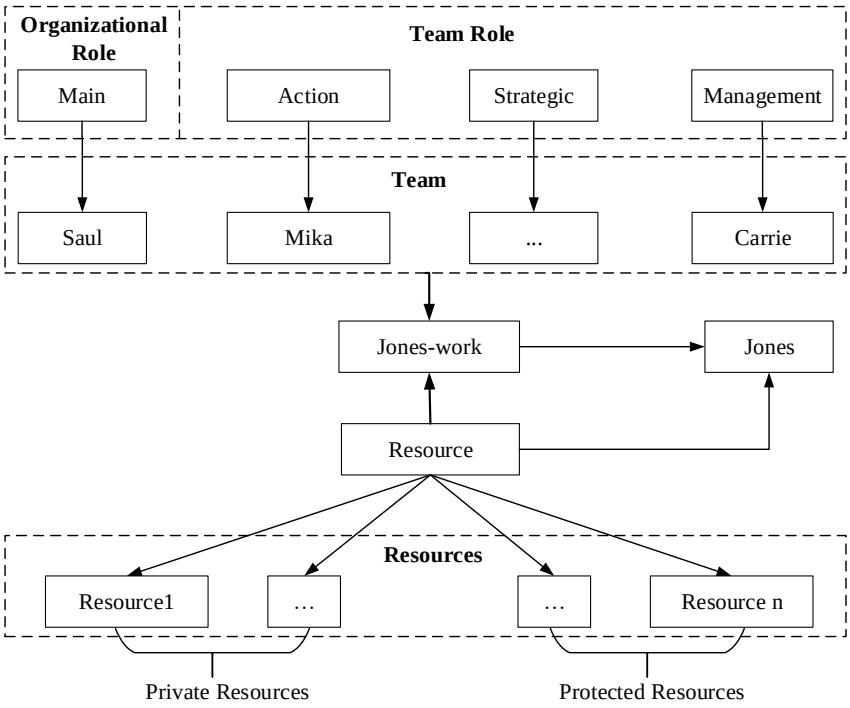


Figure 3.12: Patient *Jones* team scenario

Table 3.4: Tabular structure of policy data for *Jones*' treatment

Subject	**Role**	**Team Role**	**Object Type**	**Action**	**Permission**
Soul	Primary Doctor	Main role	*Private* and *protected*	Read/write	Permit
Mika	Dermatologist	*Action*	*Private*	Read	Permit
Mika	Dermatologist	*Action*	*protected*	Read/Write	Permit
Carrie	Medical coordinator	*Management*	*Protected*	Read	Permit

work value is set to 222 were also tested. In this case, the policy's target matches the request, but the rule's condition is not satisfied due to the non-matching work team ID. Hence, the decision for this policy is *NotApplicable*. However, since the policy is only implemented with the rules necessary for permitting access when a request is matched, PDP response *NotApplicable/Indeterminate* is interpreted as a deny response as PDP is configured as deny-based.

```
<Request xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
ReturnPolicyIdList="false">
<Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
<Attribute IncludeInResult="false"
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Mika</AttributeValue>
</Attribute>
<Attribute IncludeInResult="false"
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-Role">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Dermatologist</AttributeValue>
</Attribute>
<Attribute IncludeInResult="false"
AttributeId="urn:oasis:names:tc:xacml:1.0:work:workTeam-id">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">222</AttributeValue>
</Attribute>
</Attributes>
<Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
<Attribute IncludeInResult="false"
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-type">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">JonesPrivate</AttributeValue>
</Attribute>
</Attributes>
<Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
<Attribute IncludeInResult="false"
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Write</AttributeValue>
</Attribute>
</Attributes>
</Request>
```

Listing 3.8: XACML access request by *Mika*

```
<Response xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <Result>
 <Decision>Deny</Decision>
 <Status>
 <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
 </Status>
 <Obligations>
 </Result>
</Response>
```

Listing 3.9: Decision for request in Listing 3.8 with respect to policies in Listing 3.4

3.4 Informal Validation of WBAC

This section presents an informal validation of WBAC to ensure the model can fulfill and satisfy the main intended objectives, thus providing an access control model that strikes a balance between collaboration and safeguarding sensitive patient information. Informal validation is to examine the core functions of access control models [128] (more about formal specification and verification of WBAC core functions in chapter 4). The core functions are as follows:

1. Collaborative work initiation: The process of initiating collaborative work (discussed in section 3.2.1).

2. Policy structure: A policy is a statement of what is and what is not allowed, and the policy structure is a procedure of system policy enforcement (discussed in section 3.3.4).

3. Policy alteration for collaborative work: The WBAC model's process of altering access control policies to meet the organization and collaborative work requirements.

4. Permission alteration for collaborative work: The process of altering permissions assigned to subjects to access a resource.

5. Collaborative work termination: The process of deleting all permissions assigned to collaborative work.

3.4.1 Permission Alteration for Collaborative Work

As we mentioned earlier, permission refers to an access granted for an object and determine what a subject can do with it. Permission to access resources related to the collaborative work relies on the given team roles, which may change dynamically. For instance, suppose *Dean* answers to a compelling medical emergency that forces him to leave the country. To ensure collaborative work fluidity, he promotes *Bob* to the main role. With the main role, *Bob* has much greater control over the collaboration. The new change in *Bob* membership of the role assignment (Figure 3.13) requires no changes in role' permission. Moreover, no adjustments to existing access policies or rules are needed to grant *Bob* all permissions needed to manage *Alice* treatment. In addition, the change only affects this particular collaborative work and nothing else.

Personnel	Team Role
Bob	Action

Personnel	Team Role
Bob	Main

Figure 3.13: Role assignments alteration

3.4.2 Policy Alteration for Collaborative Work

Consider again *Alice*'s case in which *Cara* has a *thought* team role in deciding the best treatment for *Alice*. Since *Cara* does not need to see the patient face to face, she often contemplates upon decision-making from her local hospital (we assume *Cara* was invited from another hospital in the town/country). This implies that the shared resources for the *thought* team role are not accessed at the hospital where the patient receiving treatment. Observe that the first rule allows anyone with the *thought* team role to read the shared information locally (e.g., in hospital A). On the other hand, a second rule can be defined in the collaboration *PolicySet* (Listing 3.4), which allows access from a different location for a *thought* team role. Therefore, the physician with a *thought* team role can access the shared resources from other locations (e.g., hospital B). In both cases, however, only read access is given. In this case, the collaboration policy set will be modified; it is not necessary to modify any policy in the main policy set.

3.4.3 Collaborative Work Termination

Successful collaboration leads to the correct diagnosis for *Alice*. After receiving the required treatment, *Alice* is now fully recovered and has left the hospital. Collaboration between *Dean*, *Bob*, *Cara* and *Alex* is no longer needed. Subsequently, *Dean* completes the final report for *Alice* and withdraws team from the collaborative work. Now, suppose that in the future *Alex* may be inclined to review the diagnosis. He must then request access again. When the collaborative work owner revokes or withdraws the work at hand (Figure 3.14), all access to the shared resources, including those containing the patient's health or personal information, is revoked. The *workID* that is tied to the access is therefore revoked. Revocation may entail an exhaustive system search to guarantee the complete access removal for the

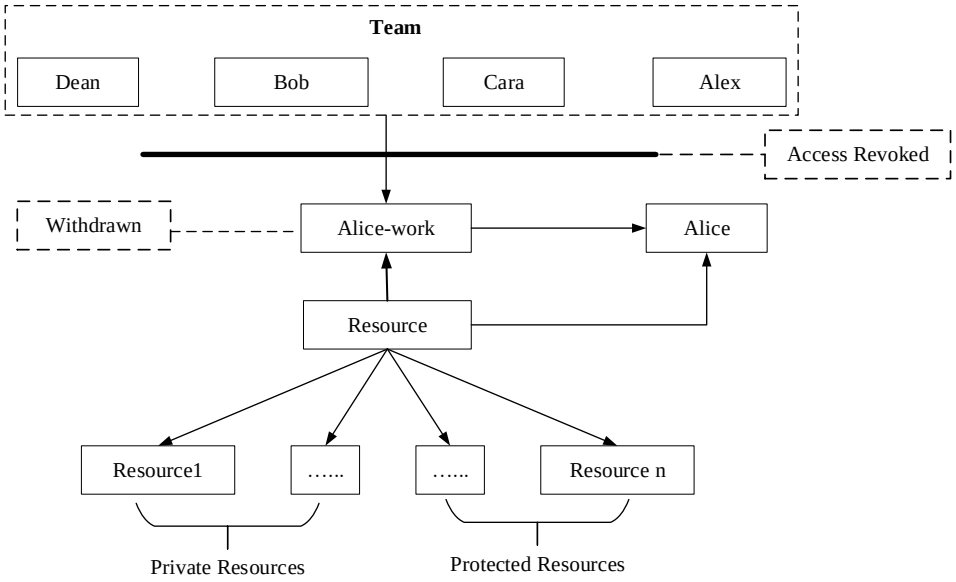


Figure 3.14: Work withdrawn to terminate collaboration

MDT members. In effect, the other collaborators will cease to have access to the work-related information. A time-stamped log entry of when a work participant entered the work flow should be made along with a corresponding time-stamp of when the work was completed/revoked. Note that, the revocation of collaborative work (e.g., *Alice* scenario, Figure 3.8) does not effect WBAC policy because revoking of collaborative work does not require any modification/changes on the access policy.

3.5 Chapter Summary

In collaborative environments such as healthcare, it is challenging to predefine all access needs based on a subject-object model. Our case scenarios give an example of such a situation, which may not be predictable. Hereby, it would be difficult to express the condition of who should join the collaboration and when a healthcare provider will request collaborative support from other parties. There are further questions to decide on the extent and limits of resource sharing. For instance, in the case of *Alice*, which sensitive data to disclose to an assisting practitioner so collaboration can be effective, and which data to hide to safeguard the patient's privacy?

We proposed WBAC to address these concerns and support the access control security and collaboration requirements (chapter 2, Section 2.3). The major contributions of the WBAC model include ensuring that access rights are adapted to the actual needs of healthcare providers and providing fine-grained control of access

rights with the *minimum necessary* standard for disclosing patient records for treatment, whereby healthcare providers obtain minimal access rights to carry out their duties.

We showed how XACML can be used to express and implement the WBAC model policy and how XACML combining algorithms can be utilized to manage the inconsistencies between different policy sets. We selected XACML because it has been proven adaptable to specifying several common access control methods, such as RBAC and ABAC. Moreover, XACML has become popular in both academia and industry as a standard for combining, maintaining and exchanging access control policies. It is an architecture for evaluating authorization requests and for issuing authorization decisions. The experiments we conducted demonstrated the applicability of XACML to supporting collaborative and distributed domains in sharing access to specific resources.

To conclude, we claim that access control models can be extended to address information sharing and information security matters. WBAC is suitable for collaborative healthcare systems. It caters to the requirements of access control in collaborative environments and provides a flexible access control model without compromising the granularity of access rights. Chapter 4 formally describes the WBAC core components, policies, and authorization constraints and evaluates the model validity.

Chapter 4

Formal Definition and Verification of WBAC

This chapter presents the formal description of the WBAC model. We start with the formal definition of core components, describe WBAC model properties and related mappings as well as present WBAC authorization constraint and WBAC policy components. Subsequently, we evaluate the validity of the security and performace of the model.

4.1 General Principles of WBAC

The WBAC includes sets of data elements called objects, operations, permissions, users, roles, team, team roles, works, healthcare organizations and sessions as shown in Figure 4.1.

- **Healthcare organization** incorporates several sectors that are dedicated and authorized to provide patient care (e.g., clinics and hospitals).
- An **object** is considered as any resource (e.g., EHRs) represented by data within the system. Access to an object potentially implies access to the information it contains [374].
- An **operation** is a term used to describe an action performed on objects (e.g., read, write, update, delete).
- **Permission** specifies the right to perform some operations on the objects for an entity (e.g., user).

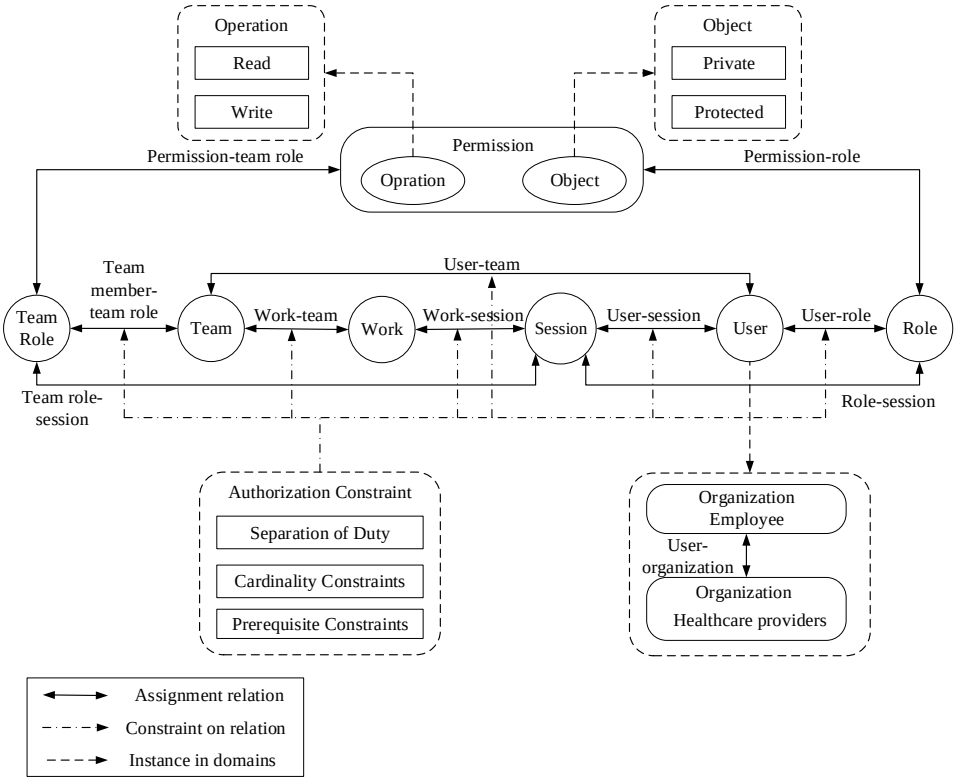


Figure 4.1: WBAC model

- A **role** is a named job function within the context of healthcare organization with some associated responsibility conferred on the user assigned to the role. In WBAC, we defined two types of roles: First is the organizational role (Section 3.1.2) where the user has a common set of permissions to perform the job function associated with the role. A user can be a member of many roles and a role can have many users. Similarly, a role can have many permissions and the permission can be assigned to many roles [317]. Second is the team role (Section 3.1.3) assigned to multiple users contributing to a team with objective of accomplishing a specific work. A member of a specific team can only be assigned to one team role. The team role of each member will restrict access permissions a member may receive. Similar to organizational role, team role is associated with a set of permissions.

- A **user** is an entity (e.g., human user, machine or process) that requires access to system objects. We consider users as human beings, practically as a healthcare professional or group of healthcare professionals that access objects in the system to perform their jobs. The permissions available to the user are the union of permissions directly assigned to the user's roles or team roles.

- A **team** encapsulates a collection of healthcare professionals in various team roles with the objective of accomplishing specific work.

- **Work** (explained in section 3.1.1). In WBAC, the treatment of *Alice*'s case (section 2.2.1) is called a *work*. This work is performed by a group of healthcare professionals who play different roles in *Alice*'s treatment and also require access to different resources in different context. We consider patient health records as objects which needed to be accessed by healthcare professionals. A work can have one or many teams working on it. Similarly, a team can be assigned to one or many works. Each team has a responsible person (team coordinator/leader) that decides who should join the team and who can perform what. A healthcare provider (team member) can have various team roles whereby each of them is tied to a different collaborative work. A team member can perform the tasks in the collaboration determined by the team role assigned to him/her in the team. Moreover, the work is associated with a state (i.e., active or inactive), and only in active state, the team assigned to this work can perform the tasks associated with the team roles. Accessing resources and performing operations related to the work is controlled by WBAC access policy.

- **Session** is an entity where a user may activate a subset of roles and team roles he/she is a member of. A user who is a member of several roles/team roles can invoke any subset of these roles/team roles that is suitable for the tasks to be accomplished in that session. The roles which are in conflict to others are declared as mutually exclusive roles [128, 151, 317]. Separation of duty either static or dynamic is a constraint that implements least privilege principle[10] on the conflicting roles [151, 215, 217]. In case of Static Separation of Duty (SSD) the user can be assigned to only one of two mutually exclusive roles and in case of Dynamic Separation of Duty (DSD) the user is not given the freedom to activate both mutually exclusive roles in the same session. The permissions available to the user are the union of permissions from all roles or team roles activated in that session.

4.2 Formal Definition of WBAC model

This section provides the formal definition of the WBAC model.

[10]The principle of least privilege requires that "a user be given no more privilege than necessary to perform a job" [126, 128].

4.2.1 Formal Definition of Core Components

The WBAC model contains the following components (discussed in section 4.1):

- *OBJ* is a set of objects; $OBJ = OBJ_A \cup OBJ_B$ where, OBJ_A is a set of *private* objects, OBJ_B is a set of *protected* objects and $OBJ_A \cap OBJ_B = \varnothing$.
- *OPR* is a set of operations.
- *PER* is a set of permissions; $PER \subseteq OBJ \times OPR$.
- *R* is a set of roles.
- *TR* is a set of team roles (discussed in Section 3.1.3).
- *USR* is a set of users.
- *T* is a set of teams, where $T \subseteq 2^{USR}$.
- *W* is a set of collaborative works; $W \subseteq 2^T \times 2^{OBJ} \times \{Active, Inactive\}$.
- *S* is a set of sessions.
- *ORG* is a set of healthcare organizations.
- *USR-ORG-A* $\subseteq$ *USR* $\times$ *ORG*: A many-to-many user to organization assignment relation such that $(usr, org) \in$ *USR-ORG-A* if and only if a user $usr \in$ *USR* is an employee at organization $org \in$ *ORG*.
- *USR-R-A* $\subseteq$ *USR* $\times$ *R*: A many-to-many role to user assignment relation such that $(usr, r) \in$ *USR-R-A* if and only if user $usr \in$ *USR* is assigned to role $r \in R$.
- *USR-T-A* $\subseteq$ *USR* $\times$ *T*: A many-to-many user to a team assignment relation such that $(usr, t) \in$ *USR-T-A* if and only if a user $usr \in$ *USR* is a member of team $t \in T$.
- *TM-TR-A* $\subseteq$ *USR* $\times$ *T* $\times$ *TR*: A many-to-many team members to a team roles assignment relation such that $(usr, t, tr) \in$ *TM-TR-A* if and only if a user $usr \in$ *USR* is a member of team $t \in T$ and holds a team role $tr \in TR$.
- *PER-R-A* $\subseteq$ *PER* $\times$ *R*: A many-to-many permission to role assignment such that $(per, r) \in$ *PER-R-A* if and only if a role $r \in R$ contains a permission $per \in PER$.

- *PER-TR-A* $\subseteq$ *PER* $\times$ *TR*: A many-to-many permission to team role assignment relation such that $(per, tr) \in$ *PER-TR-A* if and only if a team role $tr \in$ *TR* contains a permission $per \in$ *PER*.

- *T-W-A* $\subseteq$ *T* $\times$ *W*: A many-to-many team to work assignment such that $(t, w) \in$ *T-W-A* if and only if a team $t \in T$ is assigned to work $w \in W$.

- *user-session*$(usr : USR) \rightarrow 2^S$: A mapping of user *usr* onto the corresponding sessions.

- *session-user*$(s : S) \rightarrow USR$: A mapping of session *s* onto the corresponding user.

- *session-work*$(s : S) \rightarrow 2^W$: A mapping of a session *s* onto a set of works where *session-work*(s) is a set of all available works in session *s*.

4.2.2 Formal Definition of Associated Functions

This section describes access control model properties and related mappings.

- *employees*$(org : ORG) \rightarrow 2^{USR}$: A mapping of organization *org* onto a set of users. Formally: *employees*$(org) = \{usr \in USR \mid (usr, org) \in$ *USR-ORG-A*$\}$.

- *assigned-usr-role*$(r : R) \rightarrow 2^{USR}$: A mapping of role *r* onto a set of users. Formally: *assigned-usr-role*$(r) = \{usr \in USR \mid (usr, r) \in$ *USR-R-A*$\}$.

- *team-members*$(t : T) \rightarrow 2^{USR}$: A mapping of team *t* onto a set of users. Formally: *team-members*$(t) = \{usr \in USR \mid (usr, t) \in$ *USR-T-A*$\}$.

- *assigned-usr-tr*$(usr : USR, t : T) \rightarrow TR$: A mapping of team member in a specific team onto a set of team roles. Formally: *assigned-usr-tr*$(usr, t) = \{tr \in TR \mid (usr, t, tr) \in$ *TM-TR-A*$\}$.

- *teamrole-members*$(tr) = \{usr \in USR \mid \exists\, t \in T : tr \in$ *assigned-usr-tr*$(usr, t)\}$ is a set of all users assigned to team role *tr* of all teams.

- *assigned-per-role*$(r : R) \rightarrow 2^{PER}$: A mapping of role *r* onto a set of permissions. Formally: *assigned-per-role*$(r) = \{per \in PER \mid (per, r) \in$ *PER-R-A*$\}$.

- *assigned-per-teamrole*$(tr : TR) \rightarrow 2^{PER}$: A mapping of team role tr onto a set of permissions. Formally: *assigned-per-teamrole*$(tr) = \{per \in PER \mid (per, tr) \in PER\text{-}TR\text{-}A\}$.

- *assigned-team-work*$(t : T) \rightarrow 2^{W}$: A mapping of team t onto a set of works. Formally: *assigned-team-work*$(t) = \{w \in W \mid (t, w) \in T\text{-}W\text{-}A\}$.

- *session-role*$(s : S) \rightarrow 2^{R}$: A mapping of session s onto a set of roles. Formally: *session-role*$(s) = \{r \in R \mid \exists usr \in$ *session-user*$(s) \wedge (usr, r) \in USR\text{-}R\text{-}A\}$.

- *session-work-team*$(s : S, w : W) \rightarrow 2^{T}$: A mapping of work w onto a set of teams in session s. Formally: *session-work-team*$(s, w) = \{t \in T \mid w \in$ *session-work*$(s) \wedge (t, w) \in T\text{-}W\text{-}A\}$.

- *session-team-usr*$(s : S, t : T) \rightarrow 2^{USR}$: A mapping of a session s onto a set of team members in a team t. Formally: *session-team*$(s, t) = \{usr \in USR \mid \exists w \in$ *session-work*$(s) \wedge t \in$ *session-work-team*$(s, w) \wedge (usr, t) \in USR\text{-}T\text{-}A\}$.

- *session-teamrole*$(s : S, usr : USR) \rightarrow 2^{TR}$: A mapping of team members onto a set of team roles in session s. Formally: *session-teamrole*$(s, usr) = \{tr \in TR \mid \exists w \in$ *session-work*$(s) \wedge \exists t \in$ *session-work-team*$(s, w) \wedge usr \in$ *session-team-usr*$(s, t) \wedge (usr, t, tr) \in TM\text{-}TR\text{-}A\}$.

- *available-session-per*$(s : S, usr : USR) \rightarrow 2^{PER}$: The permissions available to a user in session. Formally: *available-session-per*$(s, usr) =$
$(\bigcup_{r \in session\text{-}role(s)}$ *assigned-per-role*$(r)) \cup$
$(\bigcup_{tr \in session\text{-}team\text{-}role(s,usr)}$ *assigned-per-teamrole*$(tr))$.

4.3 Formal Definition of Authorization Constraints

Authorization constraints are an important aspect mechanism for laying out higher-level organization policy [7, 8, 408]. Here, we discuss major types of WBAC authorization constraints, including variety of prerequisite constraints, separation of duty (SoD) constraints [215, 217] and cardinality constraints.

4.3.1 Prerequisite Constraints

Prerequisite constraint is based on competency and appropriateness. For example, a user can be assigned to team, only if he/she is a member of a specified organization.

C 1: *User role assignment under constraints.* It restricts the assignment of a user to a role based on his/her specialization and job function in the organization. Here if the user is a member of an organization *org* and constraint $conts(usr, org, r)$ holds, user *usr* is permitted to be assigned to role r (if needed). The constraint $conts(usr, org, r)$ restricts the ways in which the user $usr \in employees(org)$ may be assigned to role $r \in R$. Formally: let $can\text{-}assign\text{-}role : USR \times ORG \times R \rightarrow Bool$ predicate is true if the user is not assigned to role and if he/she is required to be assigned to role. Then, $\exists\, org \in ORG, \exists\, r \in R : usr \in employees(org) \wedge conts(usr, org, r) \Leftrightarrow can\text{-}assign\text{-}role(usr, org, r)$.

C 2: *User team assignment under constraints.* It restricts the assignment of a user to a team based on his/her job function in the organization and specialization. Formally: let $can\text{-}assign\text{-}team : USR \times ORG \times T \rightarrow Bool$ predicate is true if the user is not a member of the team and if he/she is required (needed) to be in the team. Then, $\exists\, org \in ORG, \exists\, r \in R, \exists\, t \in T : usr \in assigned\text{-}usr\text{-}role(r) \vee usr \in employees(org) \wedge conts(usr, org, t) \Leftrightarrow can\text{-}assign\text{-}team(usr, org, t)$. Here, if the user is member of the organization or hold a role r and constraint $conts(usr, org, t)$ holds, user *usr* is permitted to be assigned to team t (if needed). In this case, it is also considered if the user is invited from outside organization (other healthcare organization) and he/she does not hold a role in organization.

C 3: *Active work.* A collaborative work w has to be active such that team members can work on it. Formally: Let $is\text{-}active(w)$ define the state of the work w as*:* $is\text{-}active(w) = a$, where $w = (t, obj, a) \in W$. Then, $\forall\, w \in W, \forall\, s \in S : w \in session\text{-}work(s) \wedge\ is\text{-}active(w) = Active \Rightarrow session\text{-}work\text{-}team(s, w) \neq \varnothing$.

4.3.2 Separation of Duty Constraints

Separation of duty constraints are used to enforce conflict of interest policies [7, 215, 217, 343]. Conflict of interest may arise as a result of a user gaining authorization for permissions associated with conflicting roles [128]. Two major types of SoD (static and dynamic) are presented in the literature [128, 215]. Here, we discuss both of them.

I. **Static separation of duty (SSoD) constraints**: Generally place restrictions on administrative operations that have the potential to undermine higher-level organizational SoD policies. In other words, SSoD is used to enforce constraints on the assignment of users to roles or team roles to avoid a user gaining authorization for permissions associated with conflicting roles.

C 4: *Role SSoD constraints.* User-role static separation of duty (URSSoD) places constraints on the assignments of users to roles where the number of roles from URSSoD assigned to the same user cannot exceed a pre-specified number n. $URSSoD \subseteq (2^R \times N)$ is a collection of pairs (rs, n) in URSSoD, where, rs is a role set and $n \geq 2$ is an integer with the property that no user can be assigned to n or more roles from the set rs for each $(rs, n) \in URSSoD$. Formally: $\forall (rs, n) \in URSSoD, \forall\, m \subseteq rs : \mid m \mid \geq n \Rightarrow \bigcap_{r \in m} assigned\text{-}usr\text{-}role(r) = \varnothing$.

C 5: *Team SSoD constraints.* User-team static separation of duty (UTSSoD) places constraints on the assignments of users to teams. $UTSSoD \subseteq (2^T \times N)$ is a collection of pairs (ts, n) where each ts is a team set and $n \geq 2$ is an integer with the property that no user can be assigned to n or more teams from the set ts. Formally: $\forall (ts, n) \in UTSSoD, \forall\, m \subseteq ts : \mid m \mid \geq n \Rightarrow \bigcap_{t \in m} team\text{-}members(t) = \varnothing$.

C 6: *Team role SSoD constraints.* A user in one team must be assigned to exactly one team role. Formally: $\forall\, t \in T, \forall\, usr \in USR : usr \in team\text{-}members(t) \Rightarrow \mid (assigned\text{-}usr\text{-}tr(usr, t)) \mid = 1$.

II. **Dynamic separation of duty (DSoD) constraints**: Reduces the number of potential permissions that can be made available to a user by placing constraints on the roles and team roles that can be activated within or across sessions.

C 7: *Role DSoD constraints.* Session-roles dynamic separation of duty (SRDSoD) places restrictions on a user activation of roles within the same session. $SRDSoD \subseteq (2^R \times N)$ is a collection of pairs (rs, n) where each rs is a role set and $n \geq 2$ is an integer stating that no user may activate n or more roles from the set rs for each $(rs, n) \in SRDSoD$. Formally: $\forall\, s \in S, (rs, n) \in SRDSoD : \mid rs \cap session\text{-}role(s) \mid < n$.

C 8: *Team role DSoD constraints.* Session to team role assignment constraint disallows a user from activating particular team roles within the same session. Formally: $\forall\, s \in S, \forall\, usr \in USR :\mid session\text{-}teamrole(s, usr) \mid \leq 1$. It means, user can only activate one team role at a time if he/she is assigned to many teams and teams are assigned to the same work.

4.3.3 Cardinality Constraints

Cardinality constraints refer to setting a maximum number of users that can be assigned to roles and team roles. For example, a team leader role or a team coordinator role would typically be limited to a single user.

C 9: *Role cardinality constraints.* The number of authorized users for any role r can not exceed the cardinality of that role denoted as $card_{role}(r)$. Formally: $\forall\, r \in R : \mid \textit{assigned-usr-role}(r) \mid \leq card_{role}(r)$.

C 10: *Team cardinality constraints.* The number of team members for any team t can not exceed the cardinality of that team denoted as $card_{team}(t)$. Formally: $\forall\, t \in T : \mid \textit{team-members}(t) \mid \leq card_{team}(t)$.

C 11: *Team role cardinality constraints.* The number of authorized users for any team role tr can not exceed the cardinality of that role denoted as $card_{team\text{-}role}(tr)$. Formally: $\forall\, tr \in TR : \mid \textit{teamrole-members}(tr) \mid \leq card_{team\text{-}role}(tr)$.

4.4 Definition of Access Policy

This section presents the WBAC policy definition and syntax.

4.4.1 Abstract Syntax for WBAC Policy Components

The WBAC core policy structure consists of three components: *PolicySet*, *Policy* and *Rule* [226]. An abstract syntax of WBAC policies as follows:

- An attribute *Attr* is a characteristic of a user, object or operation defined as $(\textit{AttrCat}, \textit{AttrValue})$, where *AttrCat* is an attribute category and *AttrValue* is an attribute value (discussed in Section 3.3.3, Listing 3.2).
- Q is a set of an access requests. A $q \in Q$ is a tuple $(Attr_1, Attr_2, \ldots, Attr_n)$ where each $Attr_i$ is an attribute.

  Example 4.1: *Consider the request calling for an access shown in Listing 3.6 with user Dean, object AlicePrivate and operation read. Formally:*

$$\begin{aligned} q = \{&(\textit{subject-id}, \textit{Dean}),\\ &(\textit{subject-Role}, \textit{primary-doctor}),\\ &(\textit{resource-type}, \textit{AlicePrivate}),\\ &(\textit{action-id}, \textit{read})\}. \end{aligned}$$

- *AttributeDesignator* and *AttributeSelector* elements are used to retrieve a bag of attribute values (*AttrValue*) and to identify specific values in the request context, which comes from *context handler* (Figure 3.10). In this thesis, we do not directly model these elements, instead we assume the attribute values are already obtained from the attribute database (Figure 3.10).

- The *Match* element $M = (AttrCat, AttrValue)$ identifies a set of entities by matching attribute values in an *Attr* element of the request context with the embedded attribute value. Once the bag of attribute values is retrieved, function $FMatch : M \times Q \rightarrow \{Match, No\text{-}match, Indeterminate\}$ takes two arguments and returns a result of Match, no-match or indeterminate. The First argument is an embedded value, provided by the match element, and the second argument is an attribute value obtained from the access request Q which has the same category as specified in attribute category *AttrCat*.

- *AllOf* element contains a conjunctive sequence of *Match* elements; $allof(A_{id}) = \{M_1, M_2, \ldots, M_n\}$ where A_{id} is an *allof* identifier and each M_i is a sequence of *Match* elements.

- *AnyOf* element contains a disjunctive sequence of *AllOf* elements; $anyof(\varepsilon_{id}) = \{A_1, A_2, \ldots, A_n\}$ where ε_{id} is an *anyof* identifier and $\{A_1, A_2, \ldots, A_n\}$ is a disjunctive sequence of *AllOf* elements.

- *Target* is a set of targets under which the *PolicySet*, *Policy* and *Rule* is applicable. A $target = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\} \in Target$ where each ε_i is a conjunctive sequence of *AnyOf* elements.

- $policyComb_{Alg} \in \{polPermitOverRide, polDenyOverRide, polFirstApplicable, polOnlyOneApplicable\}$ and $ruleComb_{Alg} \in \{rulePermitOverRide, ruleDeny\ OverRide, ruleFirstApplicable\}$ are the procedure according to which the results of the policies and rules are combined (discussed in Section 3.3.1).

- *Rule* is a set of rules and is composed by a target, condition and an effect. A $rule_{Id} \in Rule$ is a tuple $(Target, rule_{Con}, rule_{Eff})$, where *Id* is the identification of rule, $rule_{Con}$ is a rule condition and $rule_{Eff} \in \{Permit, Deny\}$ is a rule effect that can be either a permission or denial associated with the successful evaluation of the rule.

- $rule_{Con} = \{Apply_1, Apply_2, \ldots, Apply_n\}$ is a set of *Apply* elements; $Apply_i = \{EvaRule_{Con}, \{parameters\}\}$ where $EvaRule_{Con}$ is a Boolean function over

attributes that evaluates to true or false in the *Apply* and *parameters* are inputs (attribute value and its category) to the function.

Example 4.2: *Consider the example of Rule combined with the Policy shown in Listing 3.5 with a condition to allow primary physician to process a patient's medical records that he/she is assigned to. Formally:*

$$rule_{PrimaryDoctor} = (\{\}, rule_{Con}, \{Permit\})$$

where

$$rule_{Con} = \{string\text{-}equal, \{patient\text{-}assigned\text{-}doctor, string, access\text{-}subject\} \{user\text{-}ID, string, access\text{-}subject\}\}.$$

- *Policy* is a set of policies that apply to a certain target. Its result is computed based on the chosen combining algorithm. A $policy_{Id} \in Policy$ is a tuple $(ruleComb_{Alg}, Target, Rule)$, where *Id* is the identification of policy, $ruleComb_{Alg}$ is rule combining algorithm, *Rule* is a set of rules that belongs to the *Policy*.

  Example 4.3: *Consider the Policy shown in Listing 3.5 to ensure that primary physician has a clearance to access medical records. Formally:*

$$policy_{teamManager} = (permit\text{-}overrides, \{\}, \{PrimaryDoctor\}).$$

- *PolicySet* is a set of policies sets defined by a target and a combining algorithm. A $policyset_{Id} \in PolicySet$ is a tuple $(policyComb_{Alg}, Target, Policy)$ where *Id* is the policy set identification, $policyComb_{Alg}$ is the policy combining algorithm, *Target* is a target and *Policy* is a set of policies that belongs to the *PolicySet*.

  Example 4.4: *Consider the PolicySet shown in Listing 3.3 with two policies; policy to handle access in emergency case and a policy for primary physician access. Formally (We have shorten $policy_{Id}$ and $policySet_{Id}$ for readability):*

$$policyset_{medicalRecordsAccess} = (first\text{-}applicable, \{Target\}, \{emergencyCase, teamManager\}).$$

- $RS = (Decision)$ encapsulates the the final decision produced by the PDP where *Decision* is the final decision of *PolicySet*, *Policy* and *Rule* evaluation.

4.4.2 Policy Evaluation Semantics

In the following, we present the formal semantics of WBAC policy evaluation.

4.4.2.1 Evaluation of Target

The result of evaluating the elements *Match*, *AllOf*, *AnyOf* and *Target* is one of **Match**, **No-match** or **Indeterminate**. The indeterminate value indicates that the decision of whether or not a policy is applicable cannot be determined due to an error during evaluation.

For a target evaluation, first we evaluate the *Match* elements. let *cat* be an attribute category and let $q = (Attr_1, Attr_2, \ldots, Attr_n)$ be a access request. The evaluation of the request context in order to get attribute values in q element that match with attribute category is as shown in (4.1).

$$eval_M : cat \times Q \rightarrow 2^{AV} \cup \{error\}, \text{where } AV = \{AttrValue \mid (cat, AttrValue) \in Q\}$$
$$eval_M(cat, q) = \begin{cases} \{AttrValue \mid (cat, AttrValue) \in q\} & \\ \{error\} & \text{if an error occurs} \end{cases} \tag{4.1}$$

Let $M = (C, v)$ be a *Match* where C is an attribute category and v is the embedded attribute value. Let q be a access request element. The evaluation of *Match* is as shown in 4.2 and 4.3

$$f\text{-}Equal : v \times v' \rightarrow Bool$$
$$f\text{-}Equal(v, v') = \begin{cases} \text{True} & \text{if } (v == v') \\ \text{False} & \text{otherwise} \end{cases} \tag{4.2}$$

than,

$$FMatch(M, q) = \begin{cases} \text{Match} & \text{if } eval_M(cat, q) \neq error \wedge \exists\, v' \in eval_M(cat, q) \\ & \quad : f\text{-}Equal(v, v') = True \\ \text{No-match} & \text{if } eval_M(cat, q) \neq error \wedge \exists\, v' \in eval_M(cat, q) \\ & \quad : f\text{-}Equal(v, v') = False \\ \text{Indeterminate} & \text{otherwise} \end{cases} \tag{4.3}$$

Now to evaluate the *AllOf*, *AnyOf* and *Target* element, let M be a *Match*, let A be an *AllOf*, let ε be an *AnyOf*, let *target* be a *Target* and let q be a access request. Also, suppose that $allof(A_{id}) = \{M_1, M_2, \ldots, M_n\}$ where each M_i is a *Match* element.The evaluation of *AllOf* over request $q \in Q$ is as in 4.4:

$$f\text{-}Allof : A \times Q \rightarrow \{Match, No\text{-}match, Indeterminate\}$$

$$f\text{-}Allof(A_{id}, q) = \begin{cases} \text{Match} & \text{if } \forall i, 1 \leq i \leq n : FMatch(M_i, q) = \text{Match} \\ \text{No-match} & \text{if } \exists i, 1 \leq i \leq n : FMatch(M_i, q) = \text{No-match} \\ \text{Indeterminate} & \text{otherwise} \end{cases} \tag{4.4}$$

For *AnyOf* elements, suppose that $anyof_{\varepsilon_{id}} = \{A_1, A_2, \ldots, A_n\}$ be a *AnyOf* elements and each A_i is an *AllOf* element. The evaluation of *AnyOf* over request q is as follows:

$$f\text{-}Anyof = \varepsilon \times Q \rightarrow \{Match, No\text{-}match, Indeterminate\}$$

$$f\text{-}Anyof(\varepsilon_{id}, q) = \begin{cases} \text{Match} & \text{if } \exists i, 1 \leq i \leq n : f\text{-}Allof(A_i, q) = \text{Match} \\ \text{No-match} & \text{if } \forall i, 1 \leq i \leq n : f\text{-}Allof(A_i, q) = \text{No-match} \\ \text{Indeterminate} & \text{otherwise} \end{cases} \tag{4.5}$$

Let $target = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ be a *Target* elements and each ε_i is an *AnyOf* element. The evaluation of target over request q is as shown in 4.6.

$$eval_{target} : Target \times Q \rightarrow \{Match, No\text{-}match, Indeterminate\}$$

$$eval_{target}(target, q) = \begin{cases} \text{Match} & \text{if } \forall i, 1 \leq i \leq n : f\text{-}Anyof(\varepsilon_i, q) = \text{Match} \\ \text{No-match} & \text{if } \exists i, 1 \leq i \leq n : f\text{-}Anyof(\varepsilon_i, q) = \text{No-match} \\ \text{Indeterminate} & \text{otherwise} \end{cases} \tag{4.6}$$

where an empty *Target* is always evaluated to **Match**.

4.4.2.2 Evaluation of Rule

The core part of the *Rule* is its condition $rule_{con}$. In fact, a condition can occur only within a *Rule*. If the condition evaluates to true, then the effect (permit or deny) associated with the *Rule* is returned to the parent policy. If an error occurs when a *Rule* is evaluated, the effect *Indeterminate* is returned. If none of the conditions are applicable to the request in question, a *NotApplicable* is returned.

Let $rule_{Id} = (Target, rule_{Con}, rule_{Eff})$ be a *Rule* and let q be an access request. By evaluating the *Target* element of each *Rule* as shown in 4.6, the evaluation of condition is then done as follows:

$$EvaRule_{Con} : parameters \times Q \rightarrow Bool$$

$$EvaRule_{Con}(parameters, q) = \begin{cases} \textbf{True} & \text{if } \exists\, v \in q \wedge \exists\, v' \in parameters \\ & \quad : f\text{-}Equal(v, v') = True \\ \textbf{False} & \text{otherwise} \end{cases} \tag{4.7}$$

then, the evaluation of *Rule* is determined as follows:

$$RuleEva : Rule \times Q \rightarrow \{Permit, Deny, NotApplicable, Indeterminate\}$$

$$RuleEva(rule_{Id}, q) = \begin{cases} \text{Permit} & \text{if } Condition1 \\ \text{Deny} & \text{if } Condition2 \\ \text{NotApplicable} & \text{if } Condition3 \\ \text{Indeterminate} & \text{otherwise} \end{cases} \tag{4.8}$$

where,

- Condition 1: $eval_{target}(target, q) = \text{Match} \wedge EvaRule_{Con}(parameters, q) = True \wedge rule_{Eff} = permit$.
- Condition 2: $eval_{target}(target, q) = \text{Match} \wedge EvaRule_{Con}(parameters, q) = True \wedge rule_{Eff} = deny$.
- Condition 3: $eval_{target}(target, q) = \text{No-match} \vee EvaRule_{Con}(parameters, q) = False$.

In condition 1, the rule evaluation $RuleEva(rule_{Id}, q)$ evaluates a request q to permit if the target matches (target evaluation in (4.6)) and rule condition $rule_{Con}$ evaluates to true (rule condition evaluation in (4.7)) and the rule effect $rule_{Eff}$ is permit. In

condition 2, the rule evaluation $RuleEva(rule_{Id}, q)$ evaluates a request q to deny if the target matches and rule condition $rule_{Con}$ evaluates to true and the rule effect $rule_{Eff}$ is deny. In condition 3, the rule evaluation $RuleEva(rule_{Id}, q)$ evaluates a request q to not applicable if the target does not matches or rule condition $rule_{Con}$ evaluates to false.

4.4.2.3 Evaluation of Policy

The policy evaluation is specified by the applicability of the target and the rule combining algorithms. As described in Section 3.3.2, there are three main rule combining algorithms *Deny-overrides*, *Permit-overrides* and *First-applicable*. In this section, we present the policy evaluation when the rule combining algorithms are *Permit-overrides*, *Deny-overrides* and *First-applicable* as shown in (4.9).

To formalise this, let $policy_{Id} = (ruleComb_{Alg}, Target, \{rule_1, \ldots, rule_n\})$ be a *Policy* with a set of *Rule* and let q be a request. Then, the evaluation of $policy_{Id}$ is defined as follows:

$$PoliyEva : Policy \times Q \rightarrow \{Permit, Deny, NotApplicable, Indeterminate\}$$

$$PolicyEva(policy_{Id}, q) = \begin{cases} \text{Permit} & \text{if } Case1 \vee Case2 \vee Case3 \\ \text{Deny} & \text{if } Case4 \vee Case5 \vee Case6 \\ \text{NotApplicable} & \text{if } Case7 \\ \text{Indeterminate} & \text{otherwise} \end{cases} \tag{4.9}$$

where,

- Case 1: $ruleComb_{Alg} = rulePermitOverRide \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, rule_{Id} \in Rule : RuleEva(rule_{Id}, q) = Permit.$
- Case 2: $ruleComb_{Alg} = ruleDenyOverRide \wedge eval_{target}(target, q) = \text{Match} \wedge \forall\, rule_{Id} \in Rule : RuleEva(rule_{Id}, q) = Permit.$
- Case 3: $ruleComb_{Alg} = ruleFirstApplicable \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, i \in \{1, \ldots, n\} : RuleEva(rule_i, q) = Permit \wedge \forall j \in \{1, \ldots, n\} : (j < i) \Rightarrow (rule_j = NotApplicable).$
- Case 4: $ruleComb_{Alg} = rulePermitOverRide \wedge eval_{target}(target, q) = \text{Match} \wedge \forall\, rule_{Id} \in Rule : RuleEva(rule_{Id}, q) = Deny.$
- Case 5: $ruleComb_{Alg} = ruleDenyOverRide \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, rule_{Id} \in Rule : RuleEva(rule_{Id}, q) = Deny.$

- Case 6: $ruleComb_{Alg} = ruleFirstApplicable \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, i \in \{1, \ldots, n\} : RuleEva(rule_i, q) = Deny \wedge \forall j \in \{1, \ldots, n\} : (j < i) \Rightarrow (rule_j = NotApplicable)$.

- Case 7: $ruleComb_{Alg} \in \{rulePermitOverRide, ruleDenyOverRide, ruleFirst\ Applicable\} \wedge eval_{target}(target, q) = \text{No-match} \vee \forall\, rule_{Id} \in Rule : RuleEva(rule_{Id}, q) = NotApplicable$.

In Case 1, the policy evaluation $PolicyEva(policy_{Id}, q)$ evaluates an access request to permit if the value of rule combining algorithms is equal to permit override and the target matches with the request elements (*Target* evaluation in (4.6)) and there exists a rule $rule \in Rule$ that evaluates to permit (*Rule* evaluation in (4.8)). In Case 2, the policy evaluation $PolicyEva(policy_{Id}, q)$ evaluates an access request to permit if the value of rule combining algorithms is equal to deny override and the target matches with the request elements and all rules in the set of *Rule* are evaluated to permit. In Case 3, the result of the first-applicable algorithm is described as the first *Rule* element in the sequence whose target and condition is applicable that evaluate to permit.

In Case 4, the policy evaluation $PolicyEva(policy_{Id}, q)$ evaluates an access request to deny if the value of rule combining algorithms is equal to permit override and the target matches with the request elements and there all rules in the set of *Rule* are evaluated to deny. In Case 5, the policy evaluation $PolicyEva(policy_{Id}, q)$ evaluates an access request to deny also if the value of rule combining algorithms is equal to deny override and the target matches with the request elements and there exists a rule $rule \in Rule$ that evaluated to deny. In Case 6, the result of the first-applicable algorithm is described as the first *Rule* element in the sequence whose target and condition is applicable that evaluate to deny.

In Case 7, the policy evaluation evaluates the access request to not applicable if either the target does not match with the request elements or all rules in the set of *Rule* that evaluate to not applicable with any value of combining algorithms.

4.4.2.4 Evaluation of PolicySet

Similar to policy evaluation, the *PolicySet* evaluation is specified by the applicability of the target and the value of the policy combining algorithms. As described in Section 3.3.2, there are four main policy combining algorithms. In this section, we present the policy evaluation when the rule combining algorithms are *Deny-overrides*, *Permit-overrides*, *First-applicable* and *Only-one-applicable* as shown in (4.10).

Let $policyset_{Id} = (policyComb_{Alg}, Target, \{ policy_1, \ldots, policy_n\})$ be a sequence of policy values from the *PolicySet*. We define the policy set evaluation as follows:

$$PolicySetEva : PolicySet \times Q \rightarrow \{Permit, Deny, NotApplicable, Indeterminate\}$$

$$PolicySetEva(policyset_{Id}, q) = \begin{cases} \text{Permit} & \text{if } Status1 \vee Status2 \vee Status3 \\ & \qquad \vee Status4 \\ \text{Deny} & \text{if } Status5 \vee Status6 \vee Status7 \\ & \qquad \vee Status8 \\ \text{NotApplicable} & \text{if } Status9 \\ \text{Indeterminate} & \text{otherwise} \end{cases} \tag{4.10}$$

where,

- Status 1: $policyComb_{Alg} = polPermitOverRide \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, policy_{Id} \in Policy : PolicyEva(policy_{Id}, q) = Permit$.
- Status 2: $policyComb_{Alg} = polDenyOverRide \wedge eval_{target}(target, q) = \text{Match} \wedge \forall\, policy_{Id} \in Policy : PolicyEva(policy_{Id}, q) = Permit$.
- Status 3: $policyComb_{Alg} = First\text{-}applicable \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, i \in \{1, \ldots, n\} : PolicyEva(policy_i, q) = Permit \wedge \forall\, j \in \{1, \ldots, n\} : (j < i) \Rightarrow (Policy_j = NotApplicable)$.
- Status 4: $policyComb_{Alg} = polOnlyOneApplicable \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, i \in \{1, \ldots, n\} : PolicyEva(policy_i, q) = Permit \wedge \forall\, j \in \{1, \ldots, n\} : (j \neq i) \Rightarrow (Policy_j = NotApplicable)$.
- Status 5: $policyComb_{Alg} = polPermitOverRide \wedge eval_{target}(target, q) = \text{Match} \wedge \forall\, policy_{Id} \in Policy : PolicyEva(policy_{Id}, q) = Deny$.
- Status 6: $policyComb_{Alg} = polDenyOverRide \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, policy_{Id} \in Policy : PolicyEva(policy_{Id}, q) = Deny$.
- Status 7: $policyComb_{Alg} = First\text{-}applicable \wedge eval_{target}(target, q) = \text{Match} \wedge \exists\, i \in \{1, \ldots, n\} : PolicyEva(policy_i, q) = Deny \wedge \forall\, j \in \{1, \ldots, n\} : (j < i) \Rightarrow (Policy_j = NotApplicable)$.

- Status 8: $policyComb_{Alg} = polOnlyOneApplicable \wedge eval_{target}(target, q) =$ Match $\wedge \exists i \in \{1, \ldots, n\} : PolicyEva(policy_i, q) = Deny \wedge \forall j \in \{1, \ldots, n\} : (j \neq i) \Rightarrow (Policy_j = NotApplicable)$.

- Status 9: $ruleComb_{Alg} \in \{rulePermitOverRide, ruleDenyOverRide, ruleFirstApplicable, polOnlyOneApplicable\} \wedge eval_{target}(target, q) =$ No-match $\vee \forall policy_{Id} \in Policy : PolicyEva(policy_{Id}, q) = NotApplicable$.

In permit overrides case, if any evaluation returns permit, then the result must be permit, even if other evaluations have returned deny. In deny overrides case, if any evaluation returns deny, then the result must be deny, even if other evaluations have returned permit. In the case of first-applicable, policies are evaluated in their listing order. In the case of only-one-applicable, if only one policy is considered applicable by evaluation of its target, then the result of the policy combining algorithm shall be the result of evaluating the policy.

4.4.3 WBAC Policy Management

Policy management is a sequence of modifying/updating a policy set in the access control policies. Formally: Let *policy* be a policy that needs to be added or modified and *PolicySet* is the policy set. Then,

1. If $policy \in PolicySet$, the change contains the activity of retrieving *policy* from *PolicySet* and modifying it to new policy $policy'$. Formally: $PolicySet = PolicySet \setminus \{policy\} \cup \{policy'\}$.

2. Else if $policy \notin PolicySet$, the change contains the activity of adding *policy* to *PolicySet*. Formally: $PolicySet = PolicySet \cup \{policy\}$.

4.4.4 Evaluation of WBAC Authorization

Definition 1: *Let Γ be a WBAC model as defined in Section 4.2 and 4.3 and let an access state $\gamma \in \Gamma$ contains all the information necessary to make access control decisions for a given query. Let cat be an attribute category and v be an attributes value.*

The authorization decision which constraints whether an user *usr* is able to do an operation *opr* is defined as follows:

$$f\text{-}WBAC : \Gamma \times PolicySet \times Q \rightarrow \{Permit, Deny\}$$
$$f\text{-}WBAC(\gamma, policyset_{Id}, q) = \begin{cases} \text{Permit} & \text{if } Case1 \vee Case2 \\ \text{Deny} & \text{otherwise} \end{cases} \quad (4.11)$$

where,

- Case 1: $usr, per \in \gamma \mid \exists\, q \in Q \forall\, r \in R : (usr, r) \in USR\text{-}R\text{-}A \wedge (per, r) \in PER\text{-}R\text{-}A \wedge \exists\, policyset_{Id} \in PolicySet : PolicySetEva(policyset_{Id}, q) = Permit.$

- Case 2: $usr, per \in \gamma \mid \exists\, q \in Q, \forall\, t \in T, \forall\, tr \in TR : (usr, t, tr) \in TM\text{-}TR\text{-}A \wedge (per, tr) \in PER\text{-}TR\text{-}A \wedge \exists\, policyset_{Id} \in PolicySet : PolicySetEva(policyset_{Id}, q) = Permit.$

Case 1 checks the role assigned to the requesting subject to verify whether he/she has a valid role specified in the RBAC layer (first layer, Figure 3.1). If the subject hold a role with a valid permission and there is exist a policy (third layer, Figure 3.1) to permit the request, the subject's granted. Otherwise, $f\text{-}WBAC(\gamma, policyset_{Id}, q)$ evaluates the second layer (secondary RBAC, Figure 3.1) to check if the requesting subject holds a team role with a permission and there is exist a policy to permit the request, the subject's granted.

4.5 Access Control Evaluation Algorithms

In this subsection, we present the access control evaluation algorithms.

4.5.1 PDP Evaluation Algorithm

Algorithm 1 is a PDP evaluation algorithm. It has a list of PEPs and *PolicySet*. $PDPEva(q)$ takes $q \in Q$ as an input and find the policy set applicable for the access request q (line 1). If a match is found, it calls $PolicySetEva(policyset_{Id}, q)$ to evaluate all the polices in the *PolicySet* (line 3). If the returned value is false, then the access response *RS* conveys to PEP as Indeterminate (line 4 and 5). Otherwise, the final decision *RS* would be according to *PolicySet* evaluation. PDP sends the final decision to PEP which will be enforced.

Algorithm 1 PDP evaluation algorithm ($PDPEva(q)$)

Data: PEPList and *PolicySet*
Input: $q \in Q$
Output: *RS*

1: $TargetEvaluation = eval_{target}(target, q)$ // Find the applicable *PolicySet*
2: **if** ($TargetEvaluation ==$ **Match**) **then** // Target is applicable
3: $PDPDecision = PolicySetEva(policyset_{Id}, q)$
4: **if** ($PDPDecision == false$) **then**
5: $RS = \{Indeterminate\}$
6: **else**
7: $RS = \{PDPDecision\}$
8: **end if**
9: **else**
10: NotApplicable // Target is not applicable
11: **end if**

4.5.2 Access Decision Evaluation Algorithm

The access control algorithm 2 takes an access state $\gamma \in \Gamma$, PDP, and access request q as inputs. It begins with checking if the access request is for an emergency situation (lines 1-5), then the access is granted and the security administrator will be notified for further investigation (e.g., if the access was for the purpose of patient's treatment). Otherwise, the access decision will be according to access state $\gamma \in \Gamma$ and rules defined in policy sets (lines 6-24). First, the session *s* dedicated to users is retrieved (line 6) and all active roles/team roles assigned to users in session *s* are retrieved in role array *RoleArray* (line 7). Afterwords, all available role's/team role's permission are also retrieved in the permission array *PermissionArray* (line 8-12). If the requested permission $per = (obj, opr)$ belongs to *PermissionArray* (line 13-14), algorithm 2 calls algorithm 1 for policy evaluation (line 15). The access control policies in PDP are checked. If an access rule that satisfies the access request exists , the access request will be either permitted or denied. Finally, access logs will be updated and all access decision will be recorded (line 25).

4.6 WBAC Model Security Evaluation

As discussed by *Li and Tripunitara* in [225] (security analysis in role-based access control), security evaluation answers questions such as whether an undesirable state is reachable and if every reachable state satisfies certain safety or availability properties. Examples of undesirable states include states in which (1) an unauthorized user obtains access and (2) states in which an authorized user is entitled to an ac-

Algorithm 2 Access decision evaluation algorithm

Input: $\gamma \in \Gamma$, *PDP* and $q \in Q$
Output: Access decision

1: **if** $(e.m)$ **then** // If emergency case is true
2: Grant Access
3: Inform security Administrator
4: **goto** *exit*
5: **end if**
6: $Ses = \gamma.user\text{-}session(q.usr)$
7: $RoleArray = \bigcup_{s \in Ses} session\text{-}role(s) \cup \bigcup_{s \in Ses} session\text{-}teamrole(s)$
8: **for** $(r \in RoleArray)$ **do**
9: **for** $(s \in Ses)$ **do**
10: $PermissionArray = PermissionArray \cup available\text{-}session\text{-}per(s, usr)$ // Array of all user' permissions in session
11: **end for**
12: **end for**
13: **for** $(per \in PermissionArray)$ **do**
14: **if** $((per.AttrCat = q.AttrCat)and(per.AttrValue = q.AttrValue))$ **then**
15: $policy\text{-}evaluation = PDPEva(q)$ // Call PDP evaluation algorithm
16: **if** $(policy\text{-}evaluation == permit)$ **then**
17: Grant Access
18: **goto** *exit*
19: **else**
20: Deny Access
21: Inform security Administrator
22: **end if**
23: **end if**
24: **end for**
25: **exit** Update Logs and record access decision

cess permission but does not get it. Security evaluation generalizes safety analysis as discussed in [148, 158, 211]. In this section, we evaluate the WBAC model based on the given scenario (Section 2.2.1), similar to our example of modeling structures in the XACML profile for WBAC (Section 3.3.3).

Comparable to the request model presented in Listings 3.6 and 3.8 (Sections 3.3.5 and 3.3.7), each user, object, or operation is associated with a set of attributes that may be used for access control decisions. For example, a user's attributes may include the user's role, team role, and user ID. An object's attributes may include the object type (*private* or *protected*) and object name. In our policy model (Section 3.3.3), rules are specified that are applicable to multiple-attribute requests.

4.6.1 Security Resiliency Analysis

In this subsection, we analyze the resiliency of WBAC against unauthorized access and improper access (discussed in Section 1.1.2 and 2.2.3). We begin with the definition of illegitimate accesses as:

Definition 2: *Access by subject to object is considered to be illegitimate if:*

(i) Subject (user, role, team role) or permission is not defined in the access state (Definition 1); e.g., user usr is not a member of any team role, or

(ii) policy does not permit the access.

Example 4.5: *Consider Alice's case, we assume that the initial state denoted by γ_1 is the formal model assignment as presented in Table 3.1 (For writing space we shorting medical roles as PD for primary-doctor, GP for general-practitioner, GI for Gastroenterologist and MC for medical-coordinator).*

Let $per_1 = (AlicePrivate, read)$, $per_2 = (AlicePrivate, write)$, $per_3 = (AliceProtected, read)$, $per_4 = (AliceProtected, write)$ and $per_5 = (obj_{doctorInfo}, read)$, than γ_1 is as follows:

$$
\begin{aligned}
&USR && = \{Dean, Bob, Cara, Alex\},\\
&R && = \{PD, GP, GI, MC\},\\
&TR && = \{tr_a, tr_t, tr_m\},\\
&T && = \{t_1\},\\
&W && = \{w_1\},\\
&OBJ && = \{AlicePrivate, AliceProtected\},\\
&OPR && = \{Read, Write\},\\
&PER && = \{(per_1), (per_2), (per_3), (per_4), (per_5)\},\\
&\text{Where,} &&\\
&USR\text{-}R\text{-}A && = \{(Dean, PD), (Bob, GP), (Cara, GI), (Alex, MC)\},\\
&PER\text{-}R\text{-}A && = \{(per_1, PD), (per_2, PD), (per_3, PD), (per_4, PD)\},\\
&USR\text{-}T\text{-}A && = \{(Bob, t_1), (Cara, t_1), (Alex, t_1)\},\\
&TM\text{-}TR\text{-}A && = \{((Bob, t_1), tr_a), ((Cara, t_1), tr_t), ((Alex, t_1), tr_m)\},\\
&PER\text{-}TR\text{-}A && = \{(per_1, tr_a), (per_3, tr_a), (per_3, tr_m), (per_3, tr_t), (per_5, tr_m)\},\\
&T\text{-}W\text{-}A && = \{(t_1, w_1)\}
\end{aligned}
$$

Given an access state $\gamma_1 \in \Gamma$ (Example 4.5), an access request $q \in Q$ and a policy set *PolicySet*, the security analysis takes the form $(\gamma_1, q, PolicySet)$, if subject (user, role and team role), object and operation are defined in γ_1, there exists policy in *PolicySet* are evaluated as true, then the request is either permitted or denied as described in the access decision evaluation algorithm (algorithm 2).

For our example 4.5, let consider the access request shown in listing 3.6 (Section 3.3.5) from *Dean* who has been assigned a primary doctor role and wants to read *Alice*'s *private* object *AlicePrivate*. Formally:

$$\begin{aligned} q = \{&(\textit{subject-id}, \textit{Dean}),\\ &(\textit{subject-Role}, \textit{primary-doctor}),\\ &(\textit{resource-type}, \textit{AlicePrivate}),\\ &(\textit{action-id}, \textit{read})\}. \end{aligned}$$

According to access decision evaluation algorithm (Algorithm 2), we have:

- Input: $(\gamma_1, PolicySet, q)$
  - *Emergencycase* is false (line 1-4)
  - $Ses = \{\textit{user-session}(\textit{Dean})\}$ (line 6)
  - $RoleArray = \{PD\}$ (line 8)
  - $PermissionArray = \{per_1, per_2, per_3, per_4\}$ (Line 10)
  - $(per_1.\textit{resource-type} = q.\textit{resource-type})$ and $(per_1.\textit{AlicePrivate} = q.\textit{AlicePrivate})$ is true (lines 14)
  - $(per_1.\textit{action-id} = q.\textit{action-id})$ and $(per_1.\textit{read} = q.\textit{read})$ is true (lines 14)
  - $\textit{PolicySetEvaluation} = \textit{permit}$ is ture (lines 15, 4.6 and according to our policy defined in listing 3.5)
  - Access granted (line 17)
  - Updated log and access decision recorded (line 25)
- Output: $RS = \{\textit{permit};\ \textit{status} = \textit{ok}\}$

In this scenario, the final access decision $RS = \{\textit{permit};\ \textit{status} = \textit{ok}\}$ (also shown in Listing 3.7) for the access request with respect to evaluation algorithms. Since *Dean* is included in access state γ_1 and he is assigned to primary doctor role

as well as there are rule in access policy allows the access (rule in Listing 3.5), it is noted that the access request was successfully granted because *Dean* is permitted to read *Alice*'s *private* objects. Hence, in the present of WBAC access state, illegitimate access (Definition 2) is not possible. With security we can study safety properties, availability and mutual exclusion properties. Given access state γ_1 and *PolicySet*, we can answer the questions presented in [225] as follows:

1. Safety: Let *usr* be a presumably unauthorized user. Is $usr \in USR$ possible? In other words, is there a state in which user *usr* (presumably untrusted) could be included in the user set *USR*. A "no" answer means the system is safe. In our model, we assume that state γ_1 fully determines who can perform what on what object. Also, we assume that, in addition to administrative information, *PolicySet* contains all the information about trusted users in user set *USR*. In WBAC, users are considered as the healthcare providers who, on the one hand, are fully trusted and have the authority to (unintentionally) take the system to a state that violates the security requirements, but they are trusted not to do so. On the other hand, an insider who is trusted and included in the user set $usr \in USR$ can intentionally take the system to a state that violates the security requirements. An example of such insider was given in Section 2.2.3 (Figure 2.4).

2. Availability: Let *usr* be a presumably trusted user. Is $usr \in USR$ necessary? In other words, in any state, should user *usr* be allowed to be included in user set *USR*? In WBAC, the answer should be "yes" because we want every healthcare provider to have access to resources when necessary. But we also want to ensure that every healthcare provider who has permission to access resources is included in user set *USR*.

Security analysis could ensure that the security requirements are met, as long as the users behave according to the defined policies. However, since we are dealing with insider issues (Section 2.2.3), we could assume that a user attempts to gain access to an object that is not associated with his/her privileges. The fact that access state γ_1 limits the user from accessing any object that is not associated with his/her privileges, it does not mean that the user (insider) cannot do it if he/she is motivated and has the capability to do so. However, it can be said that the security of the WBAC model is preserved as long as the users are cooperating and behaving according to the defined policies.

4.6.2 Privilege Management

Privilege management forms the basis of access control. The security of any access control model is based on how the access control policies are defined and implemented in real situations. Access control policy should precisely capture the privileges and capabilities of users, and any action should be allowed by the policy [225]. To support dynamic environments such as healthcare, privileges need to be updated in a timely manner. A change or update in the access control system causes a state change from the current access state γ_1 (definition 1) to a new access state (γ_n). Examples of state changes are adding user, deleting user, adding role, and deleting role. Corresponding to a state change, there could be a policy change. For example, if a new role (r) is added to the system, r should have a new policy or update an existing policy for it.

Example 4.6 shows how to revise (if needed) the policy set when the access state of the WBAC system changes.

Example 4.6: *Recall Alice's treatment case and assume that the primary doctor, Dean, decides to consult another physician (gastroenterologist) for second opinion. Lisa (a new gastroenterologist) is now joining Alice's treatment team* t_1 *and she will be assigned the* thought *team role* tr_t*. Then, we have a new access state change* γ_2 *(underline indicates the changes in the access state) as follows:*

$$
\begin{aligned}
&USR && = \underline{\{Dean, Bob, Cara, Alex, Lisa\}},\\
&R && = \{PD, GP, GI, MC\},\\
&TR && = \{tr_a, tr_t, tr_m\},\\
&T && = \{t_1\},\\
&W && = \{w_1\},\\
&OBJ && = \{obj_a, obj_b\},\\
&OPR && = \{Read, Write\},\\
&PER && = \{(per_1), (per_2), (per_3), (per_4), (per_5)\},\\
&\text{Where,} &&\\
&USR\text{-}R\text{-}A && = \underline{\{(Dean, PD), (Bob, GP), (Cara, GI), (Lisa, GI), (Alex, MC)\}},\\
&PER\text{-}R\text{-}A && = \{(per_1, PD), (per_2, PD), (per_3, PD), (per_4, PD)\},\\
&USR\text{-}T\text{-}A && = \underline{\{(Bob, t_1), (Cara, t_1), (Alex, t_1), (Lisa, t_1)\}},\\
&TM\text{-}TR\text{-}A && = \underline{\{((Bob, t_1), tr_a), ((Cara, t_1), tr_t), ((Lisa, t_1), tr_t)((Alex, t_1), tr_m)\}},
\end{aligned}
$$

$$\begin{aligned}
\textit{PER-TR-A} &= \{(per_1, tr_a), (per_3, tr_a), (per_3, tr_m), (per_3, tr_t), (per_5, tr_m)\},\\
\textit{T-W-A} &= \{(t_1, w_1)\}
\end{aligned}$$

Within the new access state γ_2, a new user called *Lisa* joins the team. She holds team role tr_t. Similar to *Cara*, *Lisa* will get access (read only) to *Alice*'s *protected* resources in order to perform her task. Based on the state change here, it is not necessary to modify the policy because *Lisa* was added to team role tr_t and obtained all permissions associated with tr_t.

Example 4.7: *Assume that Lisa wants to write in Alice's* protected *objects. In this case, and based on our defined rule for tr_t, Lisa does not have permission to write in Alice's* protected *objects. Currently, the system only allows the primary doctor Dean to write to Alice's* protected *objects. We do not want to assign Lisa primary doctor because she is predominantly preoccupied with diagnosing the disease, and there is no urgent need for her to know Alice's personal information and other information in Alice's* private *objects.*

To give *Lisa* permission to write in *Alice*'s obj_b, we assume *Lisa* would be assigned *evaluator* team role ($tr_{evaluator}$) (Figure 3.4). As mentioned in section 3.2.1, access to a collaborative resource can be tailored more specifically by harnessing the stipulated team roles.

Then we have a new access state γ_3 where *Lisa* will be assigned a new team role. We have (the complete γ_2 will not be repeated but the part that would be changed is shown):

$$\begin{aligned}
\textit{TR} &= \underline{\{tr_a, tr_t, tr_m, tr_{evaluator}\}},\\
\textit{PER-TR-A} &= \underline{\{(per_1, tr_a), (per_3, tr_a), (per_3, tr_m), (per_5, tr_m), (per_3, tr_t),}\\
&\quad \underline{(per_3, tr_{evaluator}), (per_4, tr_{evaluator})\}},\\
\textit{TM-TR-A} &= \underline{\{((Bob, t_1), tr_a), ((Cara, t_1), tr_t), ((Alex, t_1), tr_m), ((Lisa, t_1), tr_{evaluator})\}},
\end{aligned}$$

Now, based on the state change, we need to modify our access policies by adding a policy for the new team role $tr_{evaluator}$. In this case, it is only necessary to modify a policy in collaborative *PolicySet medical.collaborationPolicySet* (Listing 3.4) since the changes are done on the collaborative resources. In the new policy, a write operation can be performed on an object *protected* by any user with associated team role $tr_{evaluator}$. In example 4.7, the policy change entails adding a new policy

for the new team role $tr_{evaluator}$. Therefore, it becomes $PolicySet = PolicySet \cup \{policy\}$ (WBAC Policy Management, Section 4.4.3). The complexity of access control policy revision is dependent on a number of factors, such as the number of users, number of roles, and number of resources. Due to these factors, one important aspect is to guarantee policy completeness and consistency [205, 324].

Policy completeness means that the decisions the access control model should take are completely specified in the access policy. That is, there is no situation in which the system cannot reach the goal specified by the policy because it lacks an appropriate defined policy for any access request [205]. For example, let $policyset_{Id} = (policyComb_{Alg}, Target, \{p_1, \ldots, p_n\})$ be policies for team roles. It said, the *PolicySet* is incomplete if there is no policy defined for every particular team role. Detecting incompleteness in a large set of policies is cumbersome. This challenge increases when the number of users, roles, resources and environment conditions (e.g., time and location) are included in the policies [324]. The problem of incompleteness has been studied intensively in the access control community [324]. Therefore, automated mechanisms or tools are needed to assist policy administrators with detecting incompleteness and validating policy sets. However, policy incompleteness is out of the scope of this research. In our implementation (WBAC profile of XACML, Section 3.3) incompleteness is resolved by denying all access in unspecified cases.

Policy consistency means there are no contradictions or inconsistencies in a particular policy. Policies are said to be inconsistent for a specific situation when different incompatible policies are applicable [325]. Some researchers have attempted to solve the problem of inconsistency by adding special meta-rules to access control policies [323]. For example, XACML contains conflict resolution combining algorithms (e.g., deny-overrides algorithm, first-applicable algorithm, etc.) for combining rules and policies to solve a decision conflict between multiple policies (combining algorithms are discussed in Section 3.3.1).

Another important aspect of policy revision is the correctness of the policy. In the next sections, we use the model checking tool to validate the correctness and consistency of our WBAC polices.

4.6.3 Model Checking for Security Verification

As shown in section 4.6 and 4.6.2, access decisions regarding access requests are dependent on access control policies. Therefore, ensuring correct modeling and implementation of access control policies is crucial for adopting access control

mechanisms. In this section, we apply the access control policy testing (ACPT[11]) tool [180, 397] for specifying policy models and their properties to help model and implement WBAC policies correctly during policy modeling, implementation, and verification.

Policy correctness means that evaluating the implemented policy with a test input (access request) and corresponding output (access decision) is as intended [238]. On the one hand, safety (also described in section 4.6.1) means that the system satisfies the specified conditions. It is implicit that there is no violation of the constraints specified in the policy and it is assured that the system will eventually reach the desired state after taking actions in compliance with the policy [205]. The purpose of correctness and safety verification using ACPT is to determine whether the access control properties are true and to identify a state in which the properties are not true as a counterexample for the properties [174, 205, 397]. ACPT allows simulating different policies in a particular environment. The simulation environment is built from three inputs: *subject*, *resource* and *action* (operation). ACPT provides both static and dynamic verification to ensure policy correctness and safety [180].

- Static verification ensures correct policy behaviors against its properties by using the symbolic model checker (NuSMV) [89]. NuSMV checks whether user-specified properties are satisfied by the given policy model and explores its states to detect any that violate a property.

- Dynamic verification is a testing process to assure the correctness of policies implemented in a system [180]. For dynamic verifications, the ACPT runs the policies through a series of tests powered by an advanced combinatorial testing system with an automated combinatorial testing for software (ACTS) [223] to assure policy correctness.

Moreover, ACPT uses several combining algorithms to help users specify the way to handle multiple policies in the same model. Examples of combining algorithms are first applicable algorithm, deny-overrides algorithm, and permit-overrides algorithm. The evaluation strategy concerning multiple policies is important. Generally, there can be different outcomes if one searches for the first positive match or for a blocking condition. Choosing a preferred policy may help, but different evaluation strategies may nevertheless lead to different outcomes. The evaluation strategy must hence be reconciled in the overall policy.

[11] A tool of NIST SP800-192 and XACML 2.0/3.0 for access control policy composition, analysis, tests, leak inspection, and verification.

4.7 WBAC Model Specification in ACPT

The ACPT tool is used to set up an environment that contains the subject, resources and operation. The policy model and its verification property are converted into a NuSMV model, where the model is checked to see if it is satisfied. A combinatorial test for detecting insufficient rule coverage by a specified property set is also conducted. Given an access request $q \in Q$ and a rule in *PolicySet*, we say that *PolicySet* covers q if the *PolicySet* is applicable to q. Intuitively, the higher the rule coverage of a set of requests, the better the chance that specification errors will be discovered [239, 397].

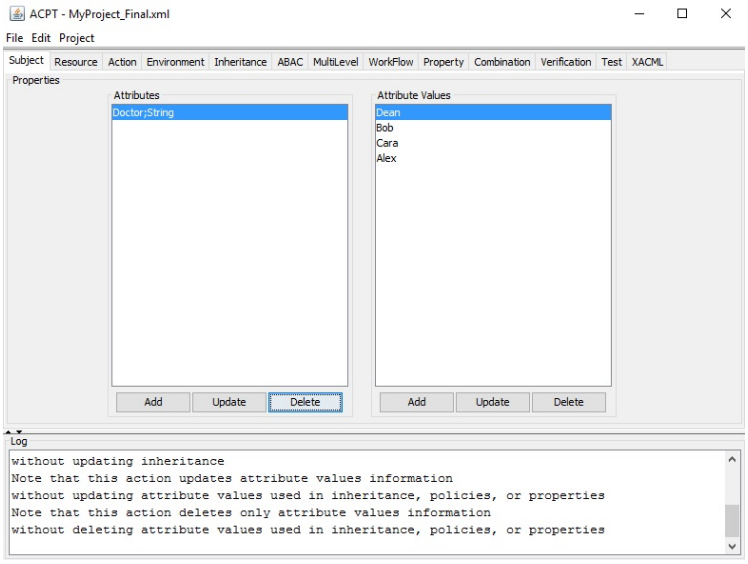
4.7.1 Modeling Structures

The subject, object and operation are represented by an independent attribute and value. Subjects, resources and operation are elements defined by a set of couples (*attribute* and *value*), for example, the *attribute* "Doctor" and its value *Dean* (Figure 4.2a), and the *attribute protected* and its value *Alice-old-medical-records* (Figure 4.2b).

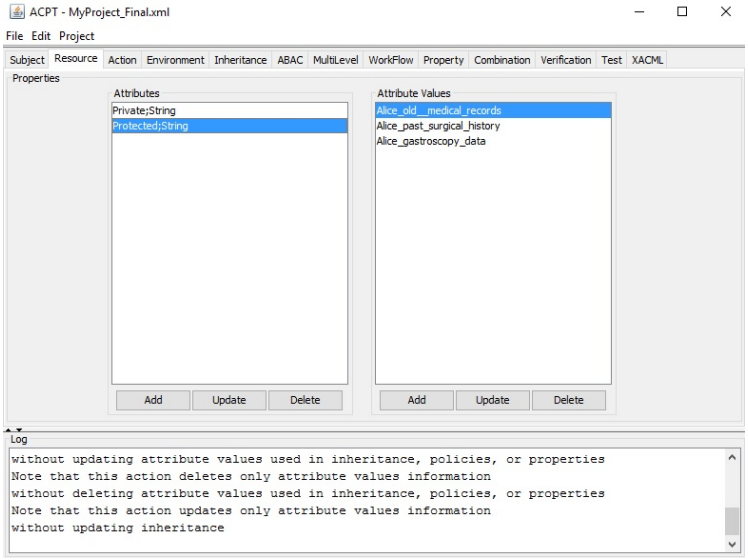
4.7.2 Verification of Properties

To ensure correct behaviors of a policy against its properties, we apply static verification and dynamic verification on the policy to verify whether its properties are satisfied.

1. **Static verification**: Let *PolicySet* be a policy set and p its property. ACPT takes *PolicySet* and p as input and verifies *PolicySet* against p using NuSMV. Figure 4.3 shows an example of properties. Figure 4.3a shows a property that describes the conditions for granting *Cara* permission to write to old medical records (*protected* object) and Figure 4.3b shows a property that describes the conditions for granting *Cara* permission to read *Alice*'s personal information (*private* object). ACPT coverts the property into NuSMV format (Figure 4.4). NuSMV takes the description of finite state systems of the model and specified properties as input. Then it verifies the finite state systems against the properties. NuSMV produces a verification report on whether the given properties are satisfied or not (Figure 4.5). If the property is violated, the model checker indicates this and provides a counterexample with a trace of parameter input values and states that will prove the property to be false.



(a) Subject specification



(b) Resource specification

Figure 4.2: Model specification and composition

Figure 4.5a represents an output of ACPT. It is evident that the verification of *PolicySet* against p is false because *Cara* is not permitted to write to *Alice*'s old medical records and she is also not permitted to read *Alice*'s personal information (Figure 4.5b) by the main and collaboration policies (Listing 3.3 and 3.4). *Cara*'s *thought* role within the team implies a rather clear indication of the access she needs. Since *Cara* is assigned to tr_t , she is only allowed to read *Alice*'s old medical records (Figure 4.6) to analyze the medical situation and suggest a possible solution.

```
SPEC (Doctor = Cara) & (Protected = Alice_old__medical_records) &
(Operation = Write) & (Work_1 = True)  -> decision =  Permit
```

(a) Property describing a condition of granting *Cara* permission to write to *Alice*'s old medical records

```
SPEC (Doctor = Cara) & (Private = Alice_personal_info) & (Operation
= Read) & (Work_1 = True)  -> decision =  Permit
```

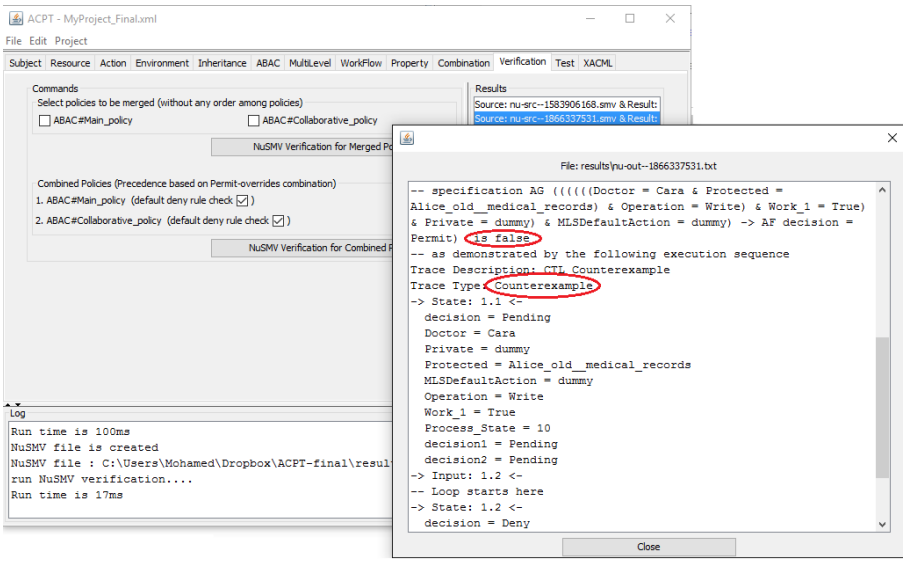
(b) Property describing a condition of granting *Cara* permission to read *Alice*'s personal information

Figure 4.3: Example properties specified in ACPT

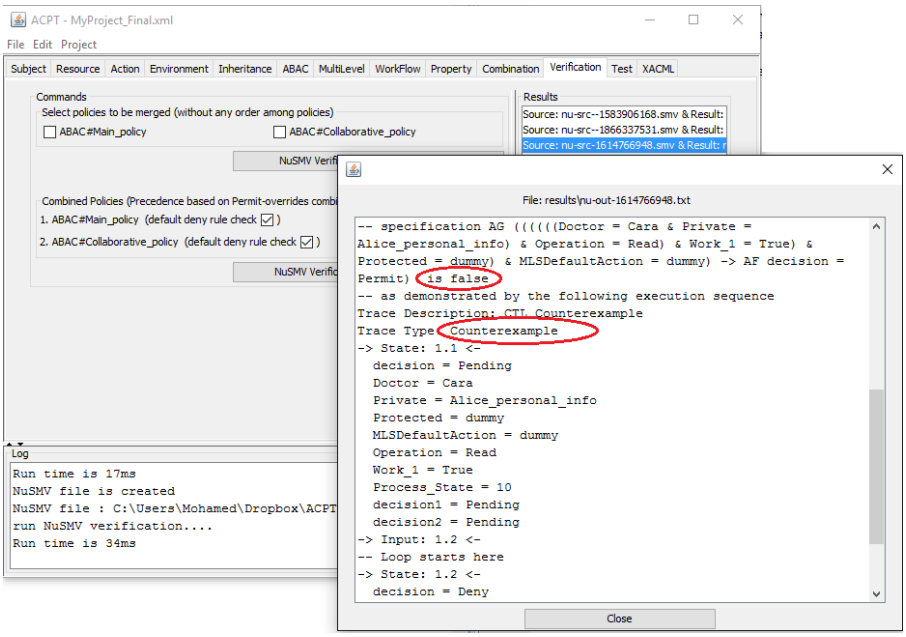
2. **Dynamic verification**: Given policy set *PolicySet* implemented in the system, dynamic verification is a process to assure the correctness of *PolicySet* through executing test inputs using combinatorial test generation. Combinatorial test generation is used to test all combinations of input parameter values. ACPT takes the description of policy content (subject, object and operation) and the attribute values as input. It then generates combinatorial tests for the given values. Figure 4.7 shows a combinatorial test output for the given subjects (*Dean*, *Bob*, *Cara* and *Alex*), resources (*private* and *protected*), and operations (read and write). The output report shows that dynamic verification of policies helps the policy authors ensure the correctness of the implemented policies by evaluating test requests and inspecting whether the evaluated de-

```
VAR
        decision: { Pending,Permit,Deny,Non-applicable};

        Doctor: { Dean,Bob,Cara,Alex};
        Private: { Alice_personal_info,Alice_private_note};
        Protected: { Alice_old__medical_records,Alice_past_surgical_history,Alice_gastroscopy_data};
        Operation: {Read,Write};
        Work_1: { True,False};

ASSIGN
 init (decision1) := Pending ;
 next (decision1) := case
                Doctor = Dean&Private = Alice_personal_info&Operation = Read:Permit;
                Doctor = Dean&Private = Alice_personal_info&Operation = Write:Permit;
                Doctor = Dean&Private = Alice_private_note&Operation = Read:Permit;

                1: Deny;
                1: Deny;
                esac;
 init (decision2) := Pending ;
 next (decision2) := case
                Doctor = Cara&Protected = Alice_old__medical_records&Operation = Read& Work_1 = True:Permit;
                Doctor = Cara&Protected = Alice_past_surgical_history&Operation = Read& Work_1 = True:Permit;
                Doctor = Cara&Protected = Alice_gastroscopy_data&Operation = Read& Work_1 = True:Permit ;
                1               : Deny;
                1               : Deny;
                esac;


        next (Doctor) :=Doctor;
        next (Private) :=Private;
        next (Protected) :=Protected;
        next (MLSDefaultAction):= MLSDefaultAction;
        next (Operation) :=Operation;
        next (Work_1) :=Work_1;


 SPEC AG ( (Doctor = Cara) & (Protected = Alice_old__medical_records) & (Operation = Write) & (Work_1 = True)
 & (Private = dummy)&(MLSDefaultAction = dummy) -> AF  decision =  Permit   )
```

Figure 4.4: NuSMV input describing an example of the model and its properties



(a) Verification results for the property presented in Figure 4.3a



(b) Verification results for the property presented in Figure 4.3b

Figure 4.5: Property verification results provided by ACPT

cisions (e.g., permit or deny) for the requests are correct. It also helps the policy authors detect insufficient policy coverage by a specified property set. If ACPT detects any missing policy, the policy author can augment the existing properties with new properties to achieve high policy coverage.

We evaluated the tested requests and inspected whether the evaluated request decisions (e.g., permit or deny) were as intended. In fact, especially in case of MDTs, inspecting all possible test inputs is not easy due to large numbers of possible test

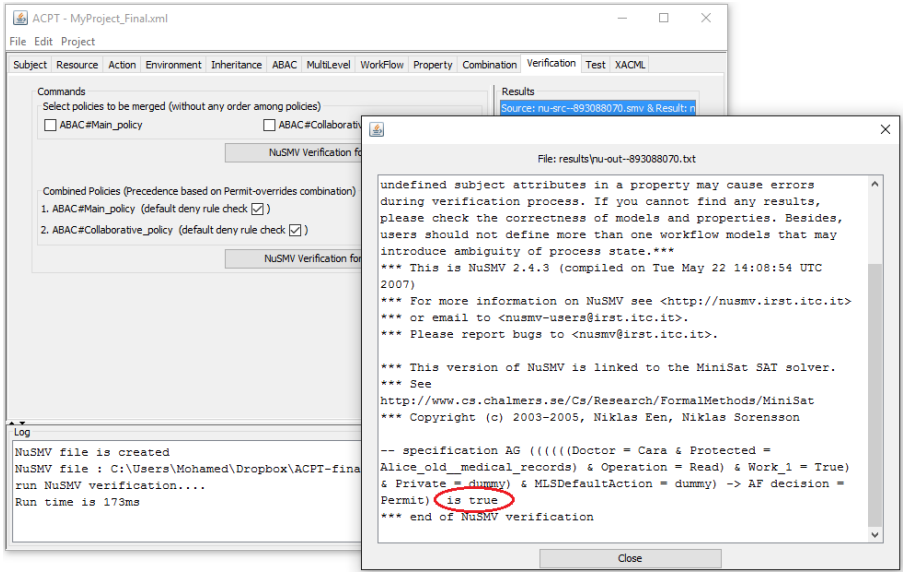


Figure 4.6: Verification results for a property describing a condition for granting *Cara* permission to read *Alice*'s old medical records

requests. Therefore, ACPT automatically generates test inputs by analyzing the policy being tested based on a given criterion (e.g., achieving high rule coverage). In addition, after conducting static verification, ACPT helps the policy authors implement XACML policies. We used XACML policy templates created by ACPT to create the XACML policy profile for the WBAC model (described in section 3.3).

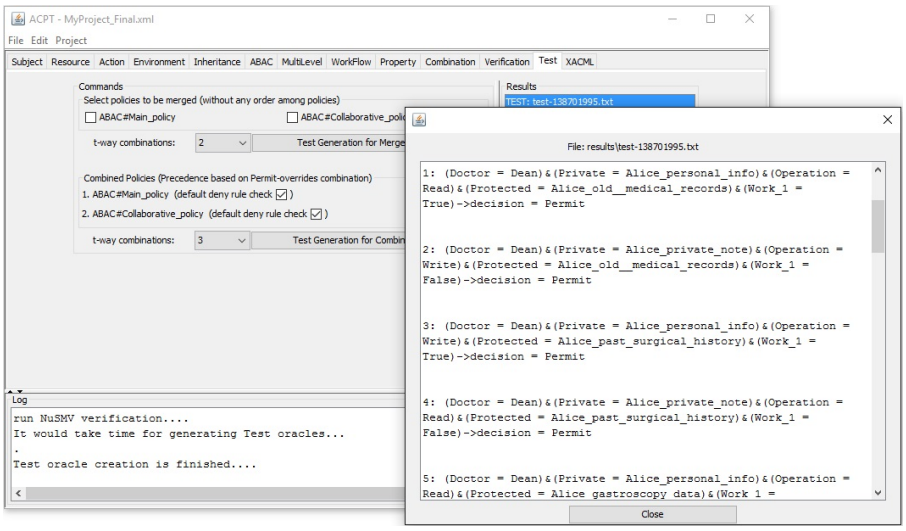


Figure 4.7: Combinatorial test for given subjects, resources, and actions by ACPT

4.8 WBAC Performance Evaluation

In this section, we conducted a number of tests in order to evaluate the performance of WBAC. We first introduce the test environment. Next, we provide evidence for the feasibility and performance of the proposed WBAC model.

4.8.1 WBAC Test Environment

We have implemented the WBAC model using *Axiomatics* PEP SDK with deployable *Axiomatics*-embedded PDP (Section 3.3.7). Our experiments were carried out on a MacBook Pro running macOS High Sierra (version 10.13) with 8 GB of memory and 2.6 GHz Intel Core $i5$ processor.

In order to get the moderate specification of the policy set (i.e., taking always the best case performance where access decision is taking early without checking all the rules), we specified (1) a policy combining algorithm to *first-applicable* (line 6, Listing 3.3 and Listing 3.4), (2) rule combining algorithm to *permit-overrides* for each policy (line 52, Listing 3.3), (3) the deny rule as the last rule in each policy and (4) rules with empty target element (in this case, the rule will inherit the target element from the policy target element in which it is contained [269]). Moreover, to maintain all assignments relation (e.g., user-to-role assignments and user-to-team assignments, etc.), we used XML file in the policy repository as shown in Listing 4.1. All the roles, team roles and other assignments that a user has are retrieved by querying this XML file. Also, for precise measurement of time intervals, we used Java method *System.nanoTime*() [189] to measure execution time. We simulated five test cases with a maximum of 50 users. We also build 10 organizational role (e.g., primary doctor and generalpractitioner, etc.) and we grant 5 different permissions per each of these 10 roles. In additional, we build our proposed team roles with 2 permissions each. We assign users to roles/team roles (example shown in Listing 4.1), and assume that there are different number of active users in sessions as shown in Table 4.1. Note that the execution time is given by millisecond (msec).

4.8.2 Performance Analysis

The processing time of our model consists of two phases. First is the pre-processing phase which consists of evaluating the roles assignment and loading granted permissions. The second is evaluation of the rules to provide the decision. We evaluate the pre-processing time of the two activities; evaluating the roles assignment

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Subjects>
    <Subject SubjectId="Dean">
        <Roles>
            <Role>primary doctor</Role>
        </Roles>
    </Subject>
    <Subject SubjectId="Bob">
    <Subject SubjectId="Cara">
    <Subject SubjectId="Alex">
    <Subject SubjectId="Dean">
        <Teams>
            <Team>111</Team>
        </Teams>
    </Subject>
    <Subject SubjectId="Bob">
        <Teams>
            <Team>111</Team>
        </Teams>
    </Subject>
</Subjects>
```

Listing 4.1: Example of user assignments relation

and loading the granted permissions, followed by the PDP processing time for rule evaluation.

Figure 4.8 represents the the experimental results (visualization of Table 4.1). The activities of sorting out the arrays (*RoleArray* and *PermissionArray*) for all active users in session took about 595 msecs to 1676 msecs as total time to evaluate all roles/team roles assigned to users and retrieve all the permission associated with the assigned roles/team roles. The analysis shows that the time grows proportionally with the input size, i.e., the growth of the number of the users and the size of the user role assignment mapping. In this case, every user is checked in the XML file (Listing 4.1) until a match is found or until all the elements have been searched. Once the arrays are sorted and matched value are found, the access request is checked against the defined rules in the policy set. We evaluate the policy evaluation process for different requests and provide the average evaluation time of executing the request.

Table 4.1: Execution time average

Teat case	Users	Operations	Execution time	Total Execution time
1	10	Evaluating the roles assignment	188	595
		Loading granted permissions	407	
2	20	Evaluating the roles assignment	354	816
		Loading granted permissions	462	
3	30	Evaluating the roles assignment	561	1109
		Loading granted permissions	548	
4	40	Evaluating the roles assignment	738	1419
		Loading granted permissions	681	
5	50	Evaluating the roles assignment	931	1676
		Loading granted permissions	745	

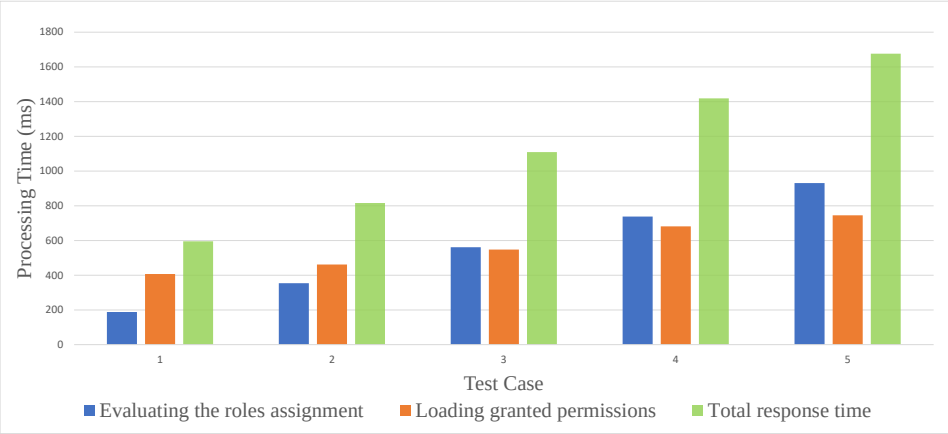
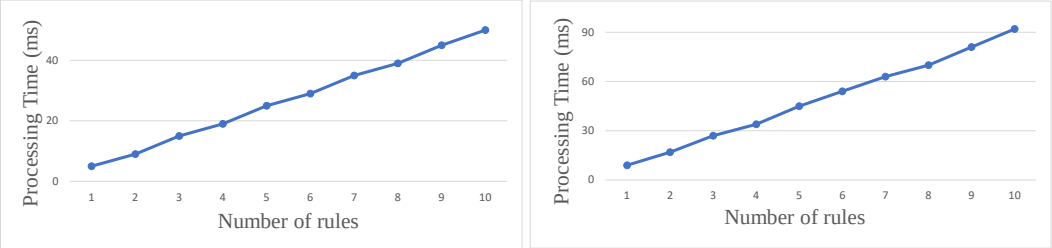


Figure 4.8: Operations execution time scale chart

Figure 4.9 depicts the actual runtime required to process single-valued and multi-valued access requests with respect to rules defined in the policy sets. Figure 4.9a shows the time required by PDP to evaluate the rules for a single-valued request (e.g., q = $\{(subject\text{-}id, Dean), (resource\text{-}type, obj_a), (action\text{-}id, read)\}$) whilst Figure 4.9b shows the average PDP processing time to evaluate the rules for multi-valued request (e.g., $q = \{(subject\text{-}id, Dean), (subject\text{-}id, Bob), (subject\text{-}id, Cara), (subject\text{-}id, Alex), (resource\text{-}type, obj_b), (action\text{-}id, read)\}$). Results show that the runtime of the XACML profile significantly increases as the size of the rule set increases. In practice and according to implementation of our policy (Section 4.8.1), the PDP does not have to evaluate the policy which does not match the target. Thus, the average time required for processing a single-valued request is almost 27 msecs and 50 msecs in the case of multi-valued request. Our performance analysis (Figure 4.8 and 4.9) indicates that the authorization time for each request is about 1150 msecs (on average), which is reasonable. Therefore, the results show that the proposed WBAC adds insignificant runtime when checking the collaboration policies (team roles, active works and collaboration rules) and is efficient and realistic to support the collaboration requirements and improve the manageability of access control during collaboration. A discussion and comparative analysis of WBAC and other models are given in Chapter 5.

4.9 Chapter Summary

This chapter formally described and discussed the WBAC model. WBAC is a promising candidate for handling collaborative work. Cooperative healthcare environments are amongst the more challenging but also serve as good testing grounds



(a) Single-valued request (b) multi-valued request

Figure 4.9: Policy evaluation time

for the applicability and practicability of WBAC.

According to the security and performance analysis of the proposed model, WBAC is suitable for collaborative healthcare systems in addressing information sharing and information security matters. It caters to the requirements of access control in collaborative environments and provides a flexible access control model without compromising the granularity of access rights. Moreover, this model is secure and easy to manage for supporting cooperative engagements that are best accomplished by organized, dynamic teams of healthcare practitioners within or among healthcare organizations whose objective is to achieve a specific work (patient treatment case). We believe WBAC meets our expectation of allowing fine-grained access control as well as enhances the practicability and manageability of access control in dynamic collaboration environments. Our conclusion is based on case evaluation. While this allows us to be optimistic about the suitability of WBAC for use within cooperative healthcare environments, we must reserve final judgment until field tests have been conducted.

Chapter 5

Specification and Validation of WBAC Authorization Constraints Using UML and OCL

The major contribution of this work lies in using Unified Modeling Language (UML) and Object Constraints Language (OCL) to validate and test WBAC authorization constraints (presented in Section 4.3). We demonstrate how the authorization constraints expressed in OCL can be implemented, tested and validated using the Eclipse Modeling Framework (EMF) tool. We additionally introduce what objects should be defined in the WBAC model, how the functionalities defined in WBAC are arranged into these objects, and how these objects work together to make access control decisions.

5.1 Background

UML and OCL are widely used in RBAC and other access control models for constraint specification and validation [8, 77, 169, 206, 216, 294, 332, 343, 345, 348]. In this section, UML, OCL and the EMF tool are briefly described.

5.1.1 Unified Modeling Language (UML)

UML [60] is a standard modeling language maintained by the *Object Management Group* (OMG) [271]. It is a collection of notations, mostly graphical, used to capture, express and build diagrams that present various views of the artifact being modeled. UML is a general-purpose visual modeling language by which it is possi-

ble to specify, visualize, and document software the system components. It captures decisions and understanding about the systems to be constructed. It also permits describing static, functional and dynamic models. UML involves several types of diagrams that are divided in three categories:

1. *Structure diagrams* emphasize what must be present in the system being modeled. An example of a structure diagram is the class diagram. A class diagram is a static model that provides a structural view of information in the system. Classes are defined in terms of their attributes and relationships, whereby the relationships include specific associations between classes.

2. *Behavior diagrams* illustrate the behavior of a system and emphasize what must happen in the system being modeled. An example is the activity diagram that describes the step-by-step business and operational activities of the system components. A case diagram is employed to identify the different types of system users and to represent a user's interaction with the system. Also, it shows the relationship between the user and the various use cases in which the user is involved.

3. *Interaction diagrams* emphasize the flow of control and data among the components of the system being modeled. For example, the sequence diagram shows how objects communicate with each other in terms of a sequence of messages.

5.1.2 Object Constraint Language (OCL)

OCL is a declarative language that describes constraints on object-oriented models [144, 381, 382]. Each OCL expression is written in the context of a specific class. It allows developers and designers to navigate class diagrams, formulate queries, and restrict class diagrams with integrity constraints [8, 216, 343, 345]. OCL aims to overcome the limitations of UML in terms of precisely specifying detailed aspects of a system design [72, 216]. Each OCL expression is written in the context of an instance of a specific type. The word *self* in an OCL expression refers to a contextual instance. The type of the context instance is written with the *context* keyword followed by the type name. The label *invariant* declares the constraint as an invariant constraint. Invariants are conditions that must be true during the system lifetime for all instances of a given type [216, 344].

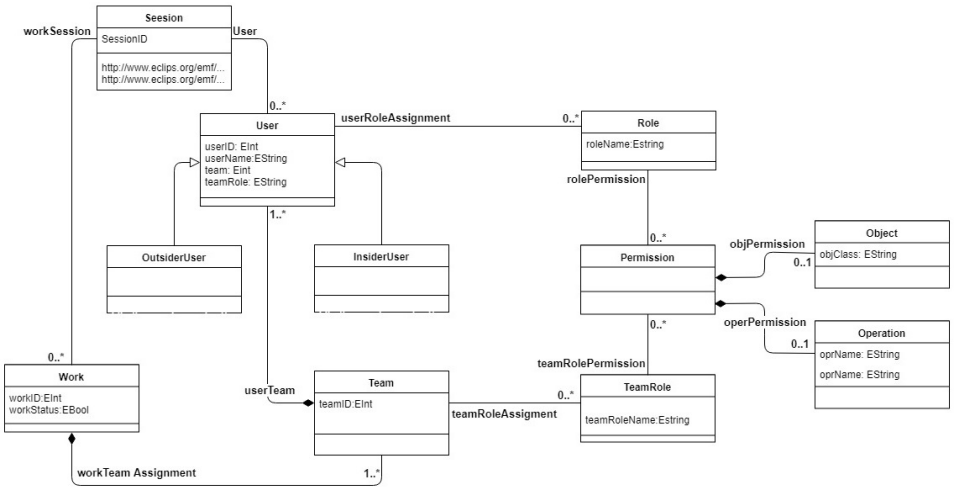


Figure 5.1: Class model for WBAC entity classes

5.1.3 Eclipse Modeling Framework (EMF)

EMF [67, 346] is a powerful, useful framework and a code generation facility for building applications based on simple model definitions. The tool supports creating UML profiles, which is one of our goals. EMF also supports OCL for defining constraints. More specifically, WBAC element sets and relations are modeled in graphical UML (Figure 5.1), while authorization constraints are specified in OCL (Listing 5.1). OCL is used to express the authorization constraints formally and precisely and EMF is applied to recognize violations of such constraints. Hence, one advantage of EMF is that it can be employed for both WBAC constraint validation and enforcement.

5.2 WBAC Specification in UML/OCL

This section presents WBAC authorization constraint specification in UML and OCL.

5.2.1 WBAC Core Classes

Figure 5.1 depicts the conceptual model of WBAC entity (presented in section 4.1), with the classes, relationships between classes, and cardinalities in the relationships. Eleven classes are defined: *user*, *insiderUser*, *outsiderUser*, *role*, *team*, *team role*, *work*, *permission*, *object*, *operation*, and *session*.

The *user class* describes users (e.g., humans, robots, computers). For simplicity and as we mentioned in previous chapters, we focus on human users (health-

```
5 class Session
6       {
7             attribute sessionID : ecore:EInt[?];
8             property users : User[*] { ordered composes  };
9       invariant uniqueUserID: self.users->isUnique(userID);
10      }
11 class User
12      {
13            attribute userID : ecore::EInt[?];
14            attribute userName : String[?];
15            attribute team : ecore::EInt[?];
16            attribute teamRole : String[?];
17      }
18      class OutsideUser extends User;
19      class InsideUser extends User
23      class permission
31      class Role
41      class TeamRole
52      class Team
75 class Work
76  {
77  attribute workID:ecore::EInt[?];
78  attribute workStatus: Boolean[?];
79  property workTeamAssgnment: Team[*] {ordered composes};
80  property worksession: Session[*] {ordered composes};
81  property workstatus: Work[*] {ordered composes};
82  }
83
84  class Object
92  class Operation
98 }
```

Listing 5.1: Example of OCL specification of a WBAC authorization constraints

care providers). The generalization relationship between the *user class* and *insiderUser* class as well as *outsiderUser* class corresponds to the inheritance between the classes. In other words, the *insiderUser* and *outsiderUser* classes are child classes of the *user class*. Insider user is an entity with an organizational role (Section 3.1.2) within the organization, while outsider user is an entity invited for collaboration (who might or might not have an organizational role within the organization). The *role class* and *teamRole class* are used to describe the users' roles and team roles. A role is a job function within the organization and team role is a job to which a team member is assigned during collaboration. The *team class* describes teams. Users invited for collaboration must be assigned to a team before they can start the work. The *work class* describes collaborative work. In case of any collaboration and team work, the *work* entity must be active. The attribute *workStatus* denotes the work state (active or inactive). The *permission class* describes permissions. The attributes *object* and *operation* hold objects from the *object class* and operations from the *operation class*.

An association and composition relationships are relationship between classes. Each relationship between classes can have a multiplicity number assigned to one or both of its ends. Multiplicity values specify in how many instances the class is involved with another class in the relationship. As shown in Figure 5.1, a user is assigned roles and teams. In practice, this means creating instances of various classes and then adding these instances to other classes, which is done with a dedicated collector method. For instance, in our diagram there is a team class that possesses a number of instances of the *insiderUser* and *outsiderUser* classes. It is important to note that instances collected by the collecting class can exist interdependent of the collecting class. For example, a user has a team role and the team role can exist without the user.

In a static model, the user role assignment, permission assignment, team assignment, user team role assignment and work assignment relations are denoted by *useRole*, *rolePermission*, *usrTeam* and *workTeamAssignment* etc, with a many-to-many cardinality. The user-session relation means a user can partake in one or more sessions to activate one or more roles/team roles in the session.

5.2.2 Constraint Specification

As shown in the WBAC class diagram (Figure 5.1), the basic entities are the *user* (insider or outsider), *role*, *team*, *team role*, *work*, *permission*, *object*, *operation* and *session* classes. Each class has an attribute that can serve as class instance identification. For example, *userID* in the user class is to identify users with a unique ID. The *userRole* and *teamRoleAssignment* relations indicate that a user can be assigned roles or team roles, respectively, and the *rolePermission* relation indicates that the permission can be assigned to a role. The *userTeam* relation indicate that the user (insider or outsider) can be assigned a team. The *workTeamAssignment* relation indicates that an active work can be assigned to the team and *teamRolePermission* indicates that a permission can be assigned to a team role.

In this section, different types of authorization constraints in OCL are specified (Listing 5.1) and validated with the help of the EMF tool. Examples are offered that demonstrate how OCL is used to specify authorization constraints. In these examples, the constraints (section 4.3) are expressed and validated with EMF.

Example 5.1: ***Prerequisite constraint***

In this example, we consider a prerequisite constraint (Section 4.3.1) stating collaborative work *w* has to be active such that team members can preform their job on it (Listing 5.2).

```
75 class Work
76  {
77  attribute workID:ecore::EInt[?];
78  attribute workStatus: Boolean[?];
79  property workTeamAssgnment: Team[*] {ordered  composes};
80  property worksession: Session[*] {ordered  composes};
81  property workstatus: Work[*] {ordered composes};
82  }
```

Listing 5.2: Example of work activation prerequisite constraint

Example 5.2: ***Separation of Duty*** *(SoD)*

As defined in Section 4.3.2, the SoD principle helps prevent fraud by identifying conflicting roles. In the WBAC model, proposed team roles are defined. A user in one team can only be assigned to exactly one team role. Mutual exclusion in terms of the *userhasteamrole* (Listing 5.3, lines 48) and *outsiderhasteamrole* (Listing 5.3, lines 49) user assignments specifies that one individual cannot have more than one team role. The constraint expression checks all users (insiders and outsiders) assigned teams and team roles and then enforces the constraint requirements.

```
52   class Team
      {
54   attribute teamID: ecore ::  EInt[?];
55   property userTeam:User[*]{ordered  composes};
56   invariant  userhasteamrole:self.userTeam.teamRoleAssignment  ->size()=1;
```

Listing 5.3: Example of OCL expression for separation of Duty

Example 5.3: ***Cardinality constraint***

This example is based on a numerical limitation on classes, which states that the number of users authorized for a team role cannot exceed the cardinality of that team role (Section 4.3.3). In this example, we consider a cardinality constraint stating that only one user (insider or outsider) can be assigned the team role *management* in a particular team. The OCL expression for this constraint is given in Listing 5.4.

```
 57   invariant oneManager:  self.userTeam.teamRoleAssignment ->
union(self.userTeam.teamRoleAssignment-  > select(teamRoleName='Manager'))->size()=1;
```

Listing 5.4: Example of OCL expression for a cardinality constraint

5.2.3 Testing and Validation

In our test and validation, we assume *Alice*'s case (presented in previous chapters). Four healthcare providers (*Dean*, *Bob*, *Cara* and *Alex*) are working on a case (patient *Alice*'s treatment). The members join the team and are assigned team roles based on the required job functions in the example of Section 4.6.

Through WBAC authorization constraint validation, conflicting constraints can be detected and missing constraints identified. Validation can be done prior to WBAC policy deployment, i.e., during the design phase. We specifically implemented all constraints explained in Section 5.2.2. The SoD constraint was implemented to prevent any of the users (*Dean*, *Bob*, *Cara* and *Alex*) from being assigned two team roles in the same team. For example, *Cara* cannot be assigned the *action* and *thought* team roles simultaneously. The EMF screenshot in Figure 5.2 displays the situation after the user (*Cara*) has been assigned two team roles in one team. Clearly, the output *"userhasteamrole"* constraint (Line 56, Listing 5.3) is violated because *Cara* should not have two team roles. Hence, the current WBAC configuration is not a correct access state according to the given constraint specification.

The validation results lead to understanding there may be an access state that does not satisfy one or more authorization constraints. This could indicate the constraints are too strong or the model is inadequate. It can also mean that the authorization constraints are too weak by allowing undesired access states. Therefore, using EMF helps find conflicting constraints as well as detect missing constraints.

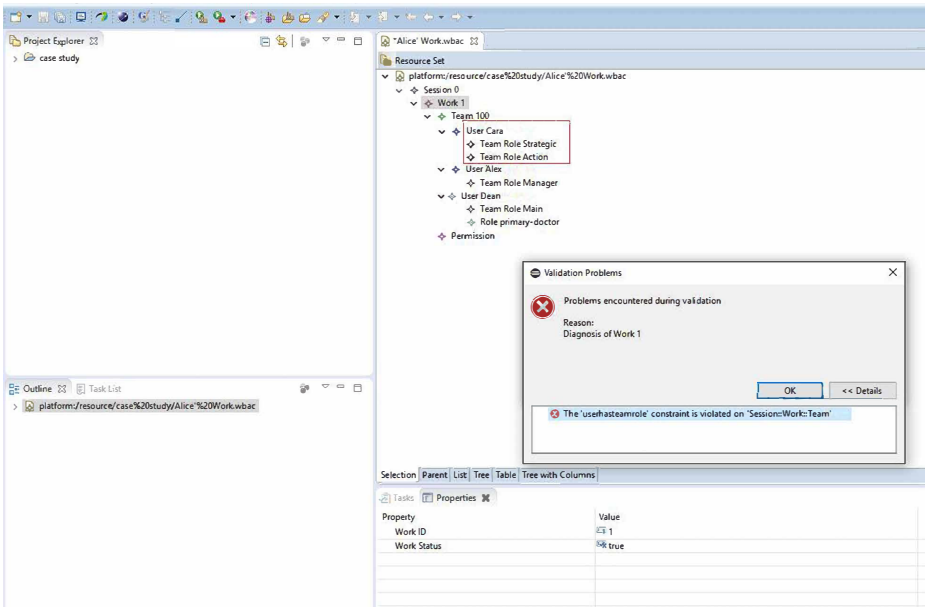


Figure 5.2: Authorization constraint enforcement in a WBAC case study

Consider *Cara*'s case in Figure 5.2, where the defined constraint *"userhasteamrole"* is evaluated to be false; here it can be said that the current WBAC configuration is incorrect for the given policy specification where a user is only allowed one team role in a given team. It is noted that more information from the constraint violation helps find the problem and resolve it. However, during testing, the problem of detecting missing constraints was difficult. For example, we assumed that we forgot to define the SoD constraint for team roles and then assigned *Cara* two team roles in one team. In this case, all defined constraints were evaluated to be true, but the access state permitted undesirable system states whereby *Cara* was allowed to be assigned mutually exclusive team roles. To solve this problem, permission review is a possible solution. Upon reviewing the current access state, we noticed that *Cara* was assigned two team roles. Therefore, it is understood that the SoD constraint must be added to the model in order to exclude such an undesirable state. Admittedly, a permission review of lager sets of policies with lager sets of users and roles is a difficult task for policy administrators. Thus, a good automated tool for checking and reviewing user permissions as well as generating an examples of access states that violate certain conditions is desirable.

5.3 WBAC Authorization Framework

As explained in Chapters 3 and 4, users obtain permissions through assigned roles and team roles in sessions. The decision algorithm (Algorithm 2) makes a decision for an access request based on the role and team role assigned to user in session as well as a set of rules. If a role *r* is assigned to a user *usr* and is activated (all authorization constraints are true), the user *usr* will get all permissions associated with the role *r*. As for team role, the permission a user will get is based on which team he/she is a member of and his/her assigend team role in that team as well as whether the collaborative work is active or not. As per chapter 4, access requests are formatted as user, operation and object, where user *usr* requests to perform operation *opr* on object *obj*. Also, as shown in the request model (Section 3.3.5), the request should contain all information (attributes) about the user, operation and object including the user's role and/or team role. WBAC enables determining if the user, once identified, is permitted to access the resource. According to Figure 5.3, WBAC is a combination of authentication and authorization processes aimed at managing and securing access to system resources while also protecting resource confidentiality and integrity, among others.

Authentication entails validating the identity establishment between two com-

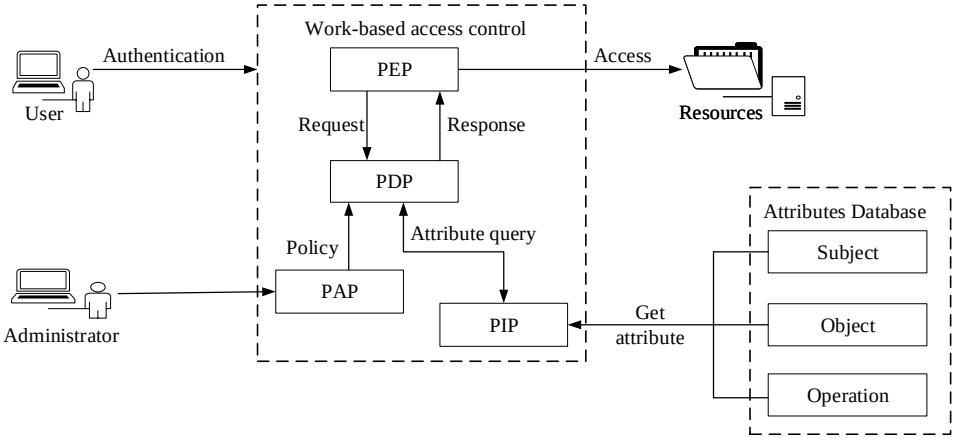


Figure 5.3: Authorization mechanism for WBAC

municating parties, showing what or who the user is. When a user requests access to a system resource, the user must first authenticate him/herself to the system. In our work [4, 5], we proposed an attribute-based authentication (ABA) scheme, which is a way to authenticate users by attributes or their properties (authentication is out of this thesis scope). Second, the WBAC authorization process decides to permit or deny the access request based on the authorization policies. PEP intercepts a user's request to access an object and then forwards the request to PDP to obtain the access decision (permit or deny). PDP receives the request from PEP and combines the user with the object information (attribute value described in Section 3.3.3), then checks if they satisfy the authorization policies. If so, the subject's access request is granted and will be enforced by PEP.

5.3.1 Evaluation Process and Decision-Making

Figure 5.4 presents a sequence diagram of the authorization evaluation process for the WBAC model. When a user sends an access request q to perform an operation on an object, PEP intercepts the call request and forwards it to PDP to check whether the user has permission to perform the requested operation on the object. The authorization system decides if the user has permission to carry out the requested operation by checking three layers: the first RBAC layer, the secondary RBAC layer and the ABAC layer (Figure 3.1).

The entire authorization process is shown in Figure 5.5. The authorization system is responsible for making an authorization decision on an access request by checking if the access request should be permitted or denied. To do so, it must interact with other model components (classes) (Figure 5.1). The access checking operation starts with gathering all attribute values in the access request (e.g., user

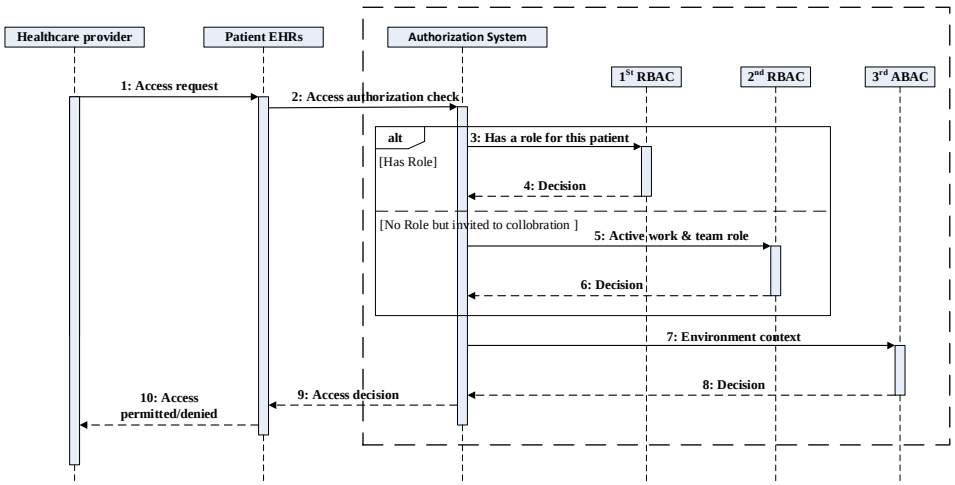


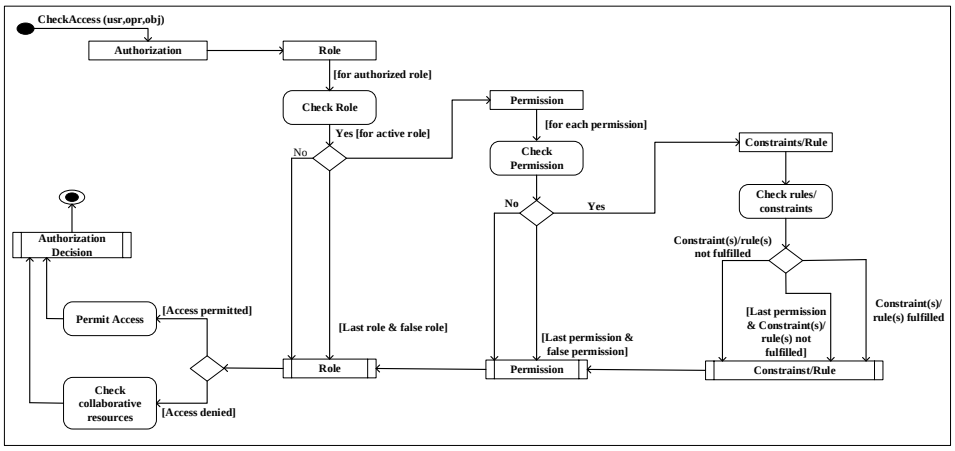
Figure 5.4: Sequence diagram of authorization process

role, object, and operation attributes) followed by checking the user's state– whether the user is in the user set. If the user is active, the checking process continues with a role check, team role check, and permission check, otherwise the checking process stops and returns the value "no".

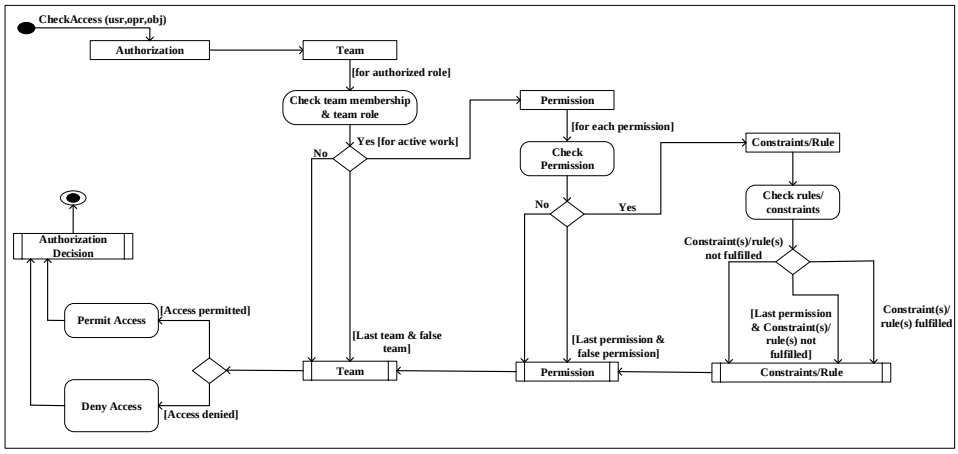
The role check process (Figure 5.5a) performs a role lookup to check if the role is assigned to the respective user. Only when the user is assigned the role, the check process continues with the permission check, otherwise it stops and returns the value "no" and the check access operation investigates the collaborative resources (Figure 5.5b). The permission lookup process checks whether the requested operation on the respective object is assigned to the corresponding role and if the input request object is equal to the permission object. If the requested operation is permitted by the role, the check access operation return "yes" and continues with the constraint and rule check on the ABAC layer. To provide a fine-grained access control, the third layer (ABAC) enforces extra constraints such as environment and context constraints. It is not sufficient to grant access only when the user holds the appropriate role.

In case the permission in the request is not assigned any role or the rule(s) check returns "no", the check access operation further investigates the collaboration policy (Figure 5.5b). The check access operation checks the user memberships in a team and if permission is granted by the team role. If the request is permitted by the respective team role and the input "requested object" is equal to a permission' object, the check access operation returns "yes" and continues with the constraint and rule check on the ABAC layer; otherwise the check process stops and the access request is denied.

Consider *Alice*'s case presented in Section 2.2.1 with four healthcare providers:



(a) Activity diagram of the role check authorization process



(b) Collaborative resources and team roles

Figure 5.5: Activity diagrams of the WBAC authorization process

Dean, *Bob*, *Cara*, *Alex*. If *Dean* sends a request to read *Alice*'s file in *Alice*'s *private* objects, the check access operation checks if the permission $(read, AlicePrivate)$ is assigned to *Dean*'s role (primary doctor). Based on the access state γ (Example 4.5), *Dean* is assigned the primary doctor role and the permission $(read, AlicePrivate)$ is assigned to the primary doctor role. Therefore, based on the role and permission checks, *Dean* is permitted to perform the operation "read" on *Alice*'s *AlicePrivate*. However, granted access based on an appropriate role is not sufficient. Thus, WBAC facilitates more fine-grained access by checking the third layer (ABAC) for additional rules, for example if *Dean* is permitted to read a file form a certain location at a particular time. In *Dean*'s case, the authorization system checks only the main policy set (Section 3.3.4), where the requesting subject is the patient's primary doctor.

If *Bob* sends a request to access *Alice*'s EHRs, the access state γ shows that *Bob* is assigned a general practitioner role, but based on the permission check, permis-

sion $(read, AlicePrivate)$ is not assigned to the general practitioner role; hence, the permission check returns "no" and the check access operation continues checking the collaboration resources (Figure 5.5b). In access state γ_1 (Example 4.5), it is assumed that *Bob* joined *Alice*'s treatment team and is assigned an *action* team role. Therefore, *Bob* is a member of t_1 and holds an *action* team role. The team check returns "yes" and the check access operation continues with permission checking. Permission $(read, AlicePrivate)$ is assigned to the *action* team role (access state γ_1), thus *Bob* is permitted to read *Alice*'s *AlicePrivate*. However, as mentioned above, all constraints associated with the corresponding request such as *work* must be true, and time and location must be fulfilled to grant *Bob* access.

In general, evaluation process complexity is dependent on the number of authorized roles assigned to a user, team membership and active works. We assume that the set of assigned roles *USR-R-A* is not a large set, but the *USR-T-A* and *T-W-A* sets could be much lager. There may be a number of active works at a hospital and many teams working on theses works. Therefore, users (healthcare providers) can join several teams to perform some duties on a work. To accelerate and improve the performance of the WBAC evaluation process, role, team, and work can be initialized and saved in the user class when a WBAC evaluation process starts.

5.3.2 Evaluation of the proposed WBAC model

Chapter 2 revealed that access control models such as RBAC, ABAC, and others are not appropriate for collaborative healthcare environments. In this section, the proposed WBAC model is compared with these approaches to better understand the differences between them with a respect to access control requirements (Section 2.3). Table 5.1 summarizes the discussion and comparative analysis of the WBAC, DAC, MAC, RBAC, ABAC, TMAC, TBAC, C-TMAC, TT-RBAC, GB-RBAC and other models.

Considering TBAC and TT-RBAC, tasks in healthcare environments usually have their own (different) characteristics and it is difficult to establish in advance access based on tasks. For instance in *Alice*'s case, it is hard to identify what task *Bob* has. In the WBAC model, as *Bob* is assigned the *action* team role, he would have all tasks related to preparing *Alice* for operation. Examples of *Bob*'s tasks are laboratory work (e.g., taking all blood tests required for the operation) and physical examination (e.g., physical examination based on gathered information related to past and current medical history, surgical history, family history, social history, use of tobacco, alcohol and illegal drugs, history of allergies, and current and recent drug therapy [403]). *Cara* is assigned the *thought* team role.

Table 5.1: Comparative analysis of the WBAC, DAC, MAC, RBAC, ABAC, TMAC, TBAC, C-TMAC, TT-RBAC, GB-RBAC and other models

Access control Models	**Requirements**									
	1	2	3	4	5	6	7	8	9	10
MAC	-	High	High	-	Static	Low	-	No	No	Complex
DAC	-	Low	Low	-	Dynamic	Low	-	No	No	Simple
RBAC	-	Medium	Low	-	Static	Low	Medium	Simple	No	Simple
ABAC	-	High	High	-	Dynamic	High	Complex	Complex	Yes	Complex
TMAC	-	Low	Low	-	Static	-	-	Simple	No	-
TBAC	-	Medium	High	-	Dynamic	-	-	-	No	-
BLAC	-	Medium	Medium	-	Dynamic	High	Complex	Complex	No	Complex
Gajanayake, et al. [134]	Yes	High	Medium	Yes	Static	Medium	Complex	Complex	Yes	Complex
Alhaqbani and Fidge [11]	Yes	High	Medium	Yes	Static	Medium	Complex	Simple	Yes	Complex
Motta and Furuie [256]	-	Low	Low	Yes	Dynamic	High	-	Simple	No	Yes
Russello, et al. [310]	-	Low	Low	Yes	Dynamic	Medium	-	-	No	Yes
Jih et al. [191]	Yes	Low	Low	No	Dynamic	Medium	Simple	Simple	No	Complex
C-TMAC [139]	-	Low	Medium	-	Dynamic	High	-	Yes	Yes	Yes
TT-RBAC [408]	-	Medium	High	-	-	Medium	-	Yes	No	Complex
GB-RBAC [227, 228]	-	High	Medium	-	Static	Medium	-	Complex	No	Yes
WBAC	Yes	High	High	Yes	Dynamic	High	Simple	Simple	Yes	Simple

Therefore, her tasks might be for example preoperative risk assessment (e.g., function of the patient's preoperative medical condition) and treatment recommendations after surgery (e.g., pain management post-op [146]). In these cases, access privileges are assigned to healthcare providers according to their team roles and not their tasks. Holding a team role would allow healthcare providers to access multiple information (based on the selective confidentiality requirement), which would allow them to work on multiple tasks related to the patient's treatment. Thus, healthcare providers assigned to the team would be permitted to access the selected objects (*necessary and relevant*) required for performing their duties.

In terms of fine-grained control, WBAC focuses on the user's role, user's team roles and target object; therefore, it can be said WBAC is classified as fine-grained access control. WBAC reduces over-privilege access arising from frequent specifications when using role in RBAC by classifying the team and objects. The level of fine-grained control of access to objects that can be authorized to healthcare providers is managed and controlled based on individual scenarios (active work, which is the patient's treatment). Although fine-grained control is very complicated in healthcare environments, WBAC's policy cab be implemented using XACML. XACML can specify rules in terms of attribute values (e.g., attributes about users, resources, actions, and the environment) that can be of various types, such as strings and integers, making WABC fine-grained.

As mentioned in previous chapters, a collaborative healthcare environment in its most basic form implies a common collaborative work undertaken by a team of healthcare providers. WBAC supports an easy means of adding, changing, manipulating, and specifying a team of users (Section 4.6). Regarding the team of users, assignment and revocation are similar to TMAC, C-TMAC and TT-RBAC, except that in WBAC, the team is categorized based on team role according to job function. Moreover, in WBAC, a team can be assigned to a collaborative work at any granularity based on the team members' team roles. In general and as explained in [253], using the concept of role in RBAC and its extension greatly reduce the management complexity of user assignment and revocation. Thus, employing the team role concept in WBAC helps solve the problem of user assignment and revocation in the case of team work.

Policy specification and policy enforcement in WBAC are the same as in RBAC. WBAC supports means of specifying and managing policies as well as using appropriate policy languages such as XACML (Section 3.3), which allows extensions or modifications in a simple and transparent manner. The main policy (Listing 3.3, Section 3.3.4) and collaboration policy (Listing 3.4, Section 3.3.4) ensure system

scalability, especially in collaborative environments, where governance policies require different organizational entities to have different responsibilities for administering various aspects of policies.

WBAC has a number of advantages including flexibility in terms of permission administration management, since roles and team roles can be updated without updating permissions for every user. It is fairly easy to assign and revoke users based on their roles and team roles. WBAC handles personalized permissions well and meets our expectation of allowing fine-grained access control, and it enhances the practicability and manageability of access control in collaboration environments.

5.3.3 Chapter Summary

This chapter demonstrated that with the help of UML and OCL several authorization constraints can be specified in the WBAC model. A number of authorization constraints were specified, including separation of duty (SOD), prerequisite and cardinality constraints. In addition, we demonstrated how the EMF tool can be employed to fulfill several practical needs, such as constraint validation, testing and configuration. Consequently, we showed how policy designers in various organizations can utilize OCL and UML in designing and/or analyzing access control systems. However, there is still room for much work on our approach.

Following a comparative analysis of the RBAC, ABAC, TMAC, TBAC, C-TMAC, TT-RBAC, GB-RBAC and WBAC models, an assessment of these models was provided based on criteria drawn from our access control requirements for collaborative environments. Based on the comparison with WBAC, it is concluded that WBAC is flexible, easy to manage and secure. It is therefore well suited to support collaborative work performed by dynamic teams in healthcare environments.

Chapter 6

Risk Assessment in the WBAC Model

This chapter presents a framework for risk assessment that extends the WBAC model by incorporating a risk assessment process and the a notion of trust the system has on its users. The framework determines the risk associated with access requests and weighting such risk against the risk appetite and risk tolerance. Specifically, an access request will be permitted if the risk tolerance outweighs the risk of granting access to information, otherwise it will be denied.

6.1 Motivation and Background

Although WBAC model seems to be a promising model for different types of MDTs' work, it cannot cope with the changing behavior of the users. As long as a user is authorized for a role/team role, the system grants him/her access if the access policy allows. As mentioned previously, WBAC is appropriate if users are well-behaved and also trusted to perform operations according to their roles/team roles and job duties. Unfortunately, evidences show that insiders do perform attacks [176, 185, 288, 296, 383]. Moreover, even if the users could be trusted, malware can be inadvertently installed and a user account can be compromised [37]. Thus, including the behavior of the users in the access control would help to monitor users who may jeopardize the information privacy and system integrity [30, 149, 193, 229]. The trust the system has on a user should be updated and adapted to suspicious changes in the user's behavior. When the user's behavior falls out of the expected pattern in a suspicious fashion, the trust the system has on him/her should be reduced. If a user is no longer trusted, the system should react by denying access to key resources.

Several researchers have recognized the advantages of adding trust to access control models [37, 50, 105, 229, 406]. In this chapter, we propose a framework that integrates WBAC with the notions of risk and trust. Risk-based WBAC makes access decisions by determining the risk associated with access requests (the user's trustworthiness level, the object's sensitivity level and the impact of the operation on the object) and weighing such risk against the risk appetite and risk tolerance. Specifically, an access request will be permitted if the risk tolerance outweighs the risk of granting access to information, otherwise it will be denied. In the following subsections, we briefly discuss basic risk assessment terminology and risk assessment approaches followed by a brief summary of related work.

6.1.1 Basic Risk Terminology

Information security risk assessment methodologies are intended to manage risk to systems. Risk factors include assets, threats, vulnerabilities, impact and countermeasures, all of which affect how the system will meet the desired outcomes [56]. Risk starts with threat events and a threat is analyzed to determine the likelihood that a threat event will occur. The likelihood of a threat occurring is assessed from the perspective of the threat actor (e.g., insider) with a defined set of capabilities, resources and motivations [3]. Moreover, the impact of threat events needs to be considered once the likelihood of threat events and vulnerabilities is understood. This is done by conducting threat modeling (e.g., see our work on threat modeling [2]) to determine what assets will be affected, the cost of the damage, and what to do about the threat. This will facilitate determining the overall risk and defining the best security controls (countermeasures) that should be put in place to protect the assets [302]. Following definitions of risk factors are based on NIST special publication 800-30 [56], Special Publication (NIST SP)-800-39 [302] and 73:2009: risk management vocabulary guide [184].

- **Asset**: Any information or resource that is of value to an organization or a person. Damage to an asset may affect normal system functionality as well as the individuals and organizations involved with the system. For simplicity, in this chapter we only consider patient EHRs as an asset.

- **Vulnerability**: A weakness in information system design, implementation or operation that could be exploited to breach asset security. Vulnerabilities are not identified only within information systems; they also can be found in organizational governance structures, business processes and external relationships (e.g., supply chains and system providers) [302].

- **Threat**: An action that takes advantage of security weaknesses (vulnerabilities) in a system and has the potential to adversely impact organizational operations and assets. The threat is analyzed to determine the likelihood that the event will occur. Threats can originate from two primary sources: humans and environmental disasters [3, 56, 96]. Here, we only consider humans threats.

- **Impact**: The level of impact from a threat event is the magnitude of harm that can be expected to damage the reputation of the organization and its productivity. The impact of threats on healthcare organizations can involve unauthorized disclosure of information (loss of confidentiality), unauthorized modification or destruction of information (loss of integrity), disruption of access to and/or use of information (loss of availability), or cross-origination data flows is not comply with relevant laws, policies and regulations (loss of compliance). These impacts are not measured well because they vary based on the insider's motivation and objectives [178]. For example, angry insider threats (e.g., actions taken in anger regarding bonuses or compensation) can have severe consequences on all organizational levels [3]. Thus, a rather small or meaningless motivation can have huge impact [284]. On the other hand, the impact may not depend on motivation. For example, an innocent act (unintentional) can have a devastating effect as a maliciously motivated attack [69, 178, 284]. The goal may therefore be to avoid catastrophic consequences regardless of the motivation.

- **Risk**: NIST [56] has defined risk as a function of the likelihood of a threat event's occurrence and potential adverse impact should the event occur. The likelihood of a threat occurring is assessed from the perspective of a threat actor (i.e., insider threat in our case) with a defined set of capabilities and resources. The likelihood of threat occurrence is a weighted risk based on an analysis of the likelihood that a given threat is capable of exploiting a given vulnerability. For humans threats, an assessment of likelihood of occurrence is typically based on: (i) the trust relationships between the system and the adversary [36, 302], (ii) adversary intent; (iii) adversary capability; and (iv) adversary targeting [56], etc.. We determine the threat likelihood based on user (i.e., healthcare provider) trust level and the impact based on the object sensitivity level and effect of the operation (e.g., read) on object.

- **Risk assessment**: A systematic process of evaluating and determining the quantitative or qualitative estimates of potential risks related to a system. The

process allows organizations to identify threats and evaluate their risks to determine an appropriate course of action (i.e., risk mitigation plan). Standard approaches and methodologies for managing risk have been developed, which include the NIST risk management methodology (NIST special publication 800-30) [56], the *International Standards Organization guidance* for information security risk management (ISO 27005) [177] and *Microsoft Security Development Lifecycle* (SDL) [170]. Despite minor terminology differences, the intent of all approaches and methodologies is the same.

- **Risk mitigation plan**: The process of developing options and treatment actions to reduce risks and enhance system security. There are a number of possible treatments once a risk has been identified as follows [2, 56, 300]:

  1. **Risk acceptance**: Where the risk is within the organization's appetite.
  2. **Risk transfer**: When a risk has to be transferred to a third party, such as purchasing insurance to cover the risk.
  3. **Risk mitigation**: When a countermeasure (e.g., use of supporting, preventive, and detective controls) is implemented to minimize the risk.
  4. **Risk avoidance**: When it is better to avoid the risk by stopping or eliminating the risk causing activity (e.g., shut down certain functions in the system when risks are identified).

6.1.2 Related Work

A number of studies on risk assessment and access control based on trust have been done over many years [53, 54, 56, 79, 85, 204, 233, 316, 378]. This subsection presents a brief summary of related work.

To determine whether user access risk levels are on par with user trust levels and object security levels, *Sandhu* [316] proposed a lattice-based access control model where a user is only allowed to access an object if the trust level of the user is higher than or equal to the security level of the object. *Cheng, et al.* [85] proposed a quantified risk-adaptive access control based on fuzzy multi-level security. The authors illustrated the concept of their approach by showing how the rationale of the Bell-LaPadula model [43] and Multi-Level Security (MLS) access control model could be used to develop a risk-adaptive access control model. The model estimates the risk based on the difference between the subject security level and the object security level. Similarly, *Ni, et al.* [263] proposed a risk-based access control systems built on fuzzy inferences. They calculate risk based on the subject label

(e.g., unclassified, classified and top secret) and the object label (e.g., unclassified, classified and secret and top secret). Their model showed that fuzzy inference is a good approach for estimating access security risks.

Li, et al. [224] present a fuzzy modeling based approach for risk-based access control in eHealth cloud. Three inputs (i.e., data sensitivity, action severity, and risk history) are modeled with fuzzy set and used to calculate the level of risk associated with healthcare information access in a cloud environment. *Ma* [233] presented a formal approach to risk assessment for RBAC systems. The basic idea of this approach is assigning a security level to each user, calculating the security level for role and then calculating the risk value of the role-user assignment relation. *Bijon, et al.* [53] discussed the difference between traditional constraint-based risk mitigation and recent quantified risk-aware approaches in RBAC, and also proposed a framework for a quantified approach to risk-aware role-based access control. We believe a user's past behavior characterizes the individual's normal behavior, and thus, is an important factor that can be used to detect illegitimate access requests of insiders. The above presented studies did not consider the past behavior of users when measuring the risk. Most of them relied on the the security level of the users/role and the sensitivity level of the permission/object as the main criteria for calculating the risk. Moreover, no clear risk boundaries are defined and the models lacked adaptive features.

Baracaldo and Joshi [37] proposed a framework that extends the RBAC model. Their framework adapts to suspicious changes in users' behavior by removing privileges when the users' trust falls below a certain threshold. This threshold is computed based on a risk assessment process that includes the risk due to inference of unauthorized information. *Sharma et al.* [327] proposed a model to estimate the risk value using functions that based on the action a user wants to perform. The risk value is computed in terms of different actions and corresponding outcomes. The outcomes and the risk probability are determined along with the level of data sensitivity. The users' previous behaviour patterns are then used to estimate the overall risk value. The estimated risk value is compared with the risk threshold to determine the access decision. Moreover, *Shaikh et al.* [326] proposed a dynamic user trust calculation model based on the past behavior of the users with particular objects. The past behavior is evaluated based on the history of reward and penalty points assigned to the user after the completion of every transaction. In their model, the old and recent history (rewards and penalty points history) have equal weights. The consequences of this is that the model may not be enable to detect small changes in the recent behavior of the user in timely manner. Therefore,

if the rewards points increase, then the user trust level increases and vice versa. This model was extended in [322] by exponentially weighted moving average approach. However, in the extended model, exponentially weighted average of past behavior was obtained recursively. Also, the authors did not enforce giving the recent events higher weights than the old events. Thus, the insider could behave according to the rules to increase his/her rewards points and reach the trust level that will allow him/her to violate the system rules.

Motivated by the work presented, we propose a model that includes risk and trust in WBAC and adapts to suspicious changes in the users' behavior. In our model, we calculate the risk associated with an access request using the user trust level, object sensitivity level, and the impact of the operation on the object. The past behavior of users is dynamically calculated based on rewards and penalty points assigned to the users. In our model, rewards and penalty points are given a different weight values based on the age of their occurrence, which allows us to weigh the recent events higher than the old ones.

6.2 The Proposed Risk Assessment Model

In this section, we present our risk model. First, we give an overview of the model. Then, we present the risk appetite (i.e., amount and type of risk that an organization is willing to pursue or retain (accept) [184]) and risk tolerance (i.e., the organization's readiness to bear the risk after risk mitigation (Section 6.1.1) in order to achieve its objectives [184]), the user trust level calculation and the impact associated with permissions, followed by risk value calculation.

6.2.1 Overview of the Model

We consider WBAC model with core components, prerequisite, cardinality and SoD constraints (presented in Chapter 4). We extend the model by adding the user trust level and the impact of permissions based on the object sensitivity level to calculate the risk of the user accessing the object. Each user (healthcare provider) is associated with a trust value that is a function of his behavior. Similarly, each object (EHR record) has a sensitivity value that specifies the level of protection required for access. A user can access an object if and only if (1) he/she is assigned to a role/team role, (2) the role/team role activation in a session does not violate any constraints (Section 4.3), (3) the risk value is less than or equal to the risk tolerance, and (4) access policies permit the access.

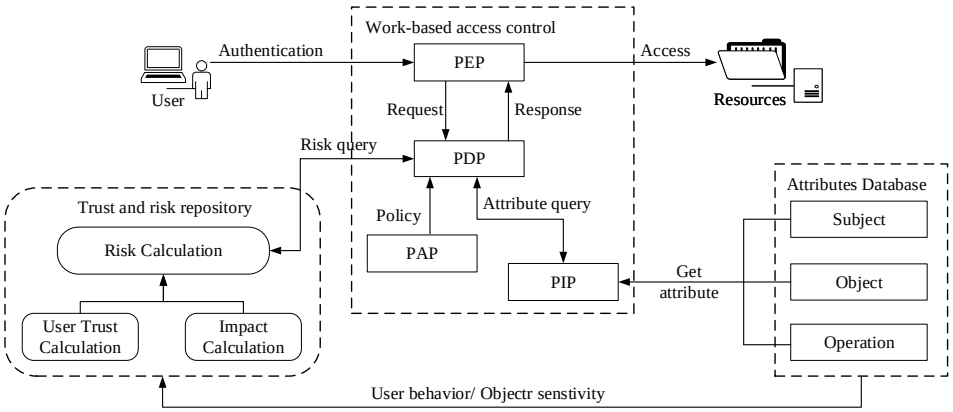


Figure 6.1: Risk-based decision model

The proposed system architecture is shown in Figure 6.1. Similar to Figure 5.3, the access requests are intercepted by the PEP, which sends them to the PDP (presented in Section 3.3.1). The PDP evaluates the policy according to the trust the system has on the user and the risk value. The trust and risk repository is used to monitor user behavior and calculate the risk value of each access request based on the trust level of the user and the impact of the requested operation on the object. The input risk factors (user behavior, object sensitivity and operation severity) are used to estimate the security risk value associated with each access request. The final risk value is then compared with the risk appetite and risk tolerance to make the final access decision.

6.2.2 Risk Appetite and Tolerance

In the WBAC, the decision (either permit or deny) would be replaced by a dynamic access decision based on the risk value, risk appetite and risk tolerance.

Definition 3: *The risk appetite value denoted by risk-appetite*(obj, opr) *is defined in the interval* $[0, 1]$ *and is the amount of risk that an organization is willing to take in order to meet their strategic objectives.*

Definition 4: *The risk tolerance value denoted by risk-tolerance*(obj, opr) *is defined in the interval* $[0, 1]$ *and is the amount of risk that an organization is willing to withstand for achieving a specific objective by having the right resources and security controls in place to tolerate a given risk.*

Risk appetite and tolerance are generally set by the board and/or executive management and are linked with the organization's strategy [300]. Figure 6.2 presents the risk scale of the WBAC model, where the risk curve is divided into three bands as follows:

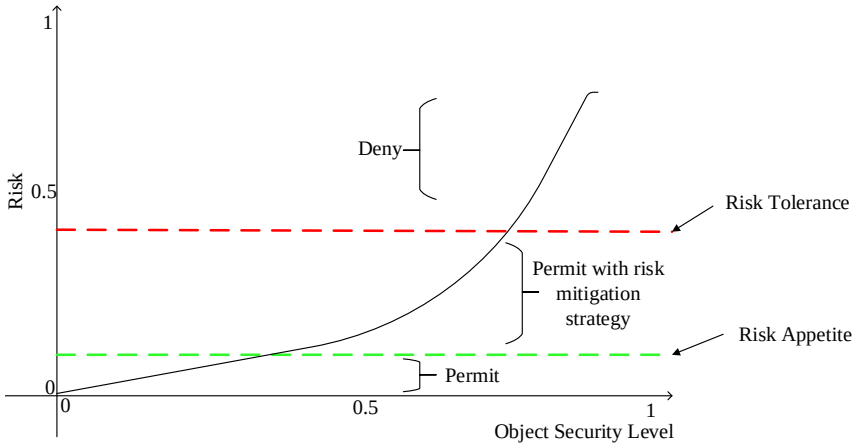


Figure 6.2: WBAC risk scale

- The first band (risk appetite boundary) is assigned with a decision *Permit* because there is no risk or the risk is very low and an entity (e.g., organization and individual) is willing to take the risk in anticipation of a rewards.

- The second band (risk tolerance boundary) is assigned a decision *Permit with risk mitigation strategy*, which are actions such as increased auditing etc. (mitigation plans are discussed in Section 6.1.1). Below the risk tolerance boundary (between the risk appetite boundary and risk tolerance boundary) is the amount of risk that an entity is willing to withstand. Above the risk tolerance boundary, the entity will not tolerate the risk. Risk tolerance varies from one entity to another depending on the risk level an entity is willing to take.

- The third band is assigned with the decision *Deny* because the risk is too high. In this case, it is not desirable to prevent a healthcare provider from accessing an object as this could cause greater damage to the patient than the risk of accessing the patient's records. This is due to requirement of "nothing must interfere with the delivery of care" [313]. But the object owner or system administrator (e.g. security administrator) must be notified of the risk and the access request will require evaluating the patient's (object owner) consent. Therefore, the problem of low detectability of data breaches (discussed in Section 2.2.3) can be examined and the object owner (e.g., patient) or system administrator may carry out an investigation and discover the main purposes of access.

As said, in case the risk value is greater than the risk appetite and risk tolerance, the healthcare provider's access request to the object should not be denied completely.

Because denying healthcare provider to access an object might cause a great harm to a patient (potentially life-threatening). However, an additional mitigation plan can be put in place to reduce the risk. An example of a mitigation plan is an additional evaluation layer such as a purpose-based access control policy [377]. The purpose of information access could be associated with the access request to specify the intention of the access request [279, 352, 400].

6.2.3 User Trust Level

User trust level has been defined by *Mayer et al.* [243] as a function of trustee's perceived ability, benevolence and integrity of the trustor's propensity to trust. In this study, we assume that each user (healthcare provider) is assigned with a trust value that represents the level of user clearance by the organization (e.g., hospital and clinic) that owns the object (EHRs) in accordance with well-established and clearly written rules, regulations, and legal principles.

Definition 5: *The trust for a user usr is denoted by* $TL(usr)$ *and is defined in the interval* $[0, 1]$*, where 1 means the user is fully trusted and 0 means the user is totally untrusted.*

User trust level may be assigned either statically or dynamically. Static assignment refers to static user attributes. For example, the trustworthiness level of a cardiologists in the cardiology department may be higher than that of the nurses who work in the same department. Dynamic assignment based on the user's access history [85], behavior [37, 322, 326] and the user reputation [53]. As we mentioned earlier (Section 6.1.2), *Shaikh et al.* [322, 326] proposed a dynamic user trust calculation model based on rewards and penalty points. The authors assumed that, on the one hand, if obligations associated with an access request to an object is successfully fulfilled, the obligation server will assign rewards points to the user with a respect to the object. On the other hand, if the obligations are not successfully fulfilled, the obligation server will assign penalty points. An example of rewards and penalty is that, if a healthcare provider accesses an object for the purpose of treatment and no misuse was reported in system, he/she would be granted a reward. If misuse was reported, then the healthcare provider would be given a penalty.

In this thesis, we modify the proposed approach by assigning different weights to rewards and penalty histories. That is, the older the action that lead to a specific reward/penalty, the smaller the weight given to that reward/penalty.

6.2.4 User Trust Calculation

For each user, we calculate a trust level value $TL(usr)$ with a respect to his/her reward and penalty events.

Definition 6: *Given a user usr and his/her events history* $RH(usr) = \{e_1, e_2, \ldots, e_n\}$ *(a set of rewards events) and* $PH(usr) = \{e_1, e_2, \ldots, e_m\}$ *(a set of penalties events), the calculation of user trust level is done based on the user history of rewards* $RE(usr, e_i)$ *and penalties* $PE(usr, e_i)$ *points with a regards to the time* τ *as follows (6.1):*

$$TL(usr) = \begin{cases} \max\left\{ \left(\dfrac{\sum\limits_{e \in RH(usr)} RE(usr,e) \cdot \lambda_{RE}^{\tau-\tau(e)} - \sum\limits_{e \in PH(usr)} PE(usr,e) \cdot \lambda_{PE}^{\tau-\tau(e)}}{\mid RH(usr) \mid + \mid PH(usr) \mid} \right), TL_{min} \right\} & \text{If history available} \\ TL_{initial} & \text{Otherwise} \end{cases} \tag{6.1}$$

where, $TL_{initial}$ is the initial trust level value of a user, TL_{min} is minimum trust level value, $0 \leq \lambda_{RE} \leq 1$ is rewards forgetting factor, $0 \leq \lambda_{PE} \leq 1$ is penalty forgetting factor, $\tau(e)$ is a time of the event e occurrence and τ is the current time of calculating the user trust level. Setting $\lambda_{RE} = 0$ means we are only considering the last reward and ignoring old rewards history. Setting $\lambda_{RE} = 1$ means we have the same weight for all rewards in the history. The case is similar to λ_{PE}.

According to (6.1), the trust level $TL(usr)$ is calculated as: (1) in case of a new user or if neither penalties nor rewards are available, the user is assigned to $TL_{initial}$, (2) if only penalties are available, the user assigned to TL_{min} where $0 < TL_{min} < TL_{initial}$, (3) if only rewards are available the trust level value will increase but never exceeds value 1 and (4) in the case where both penalties and rewards are available, the trust level value is bound between TL_{min} and 1.

The $TL_{initial}$ value can be set by the system owner (i.e., an individual or organization responsible for the overall procurement, development, integration, modification, operation and maintenance of an information system) and it should be always greater than the TL_{min} because we need to distinguish between a new user (without history) and a user who behaving badly (with more penalties events) in the system. Healthcare providers are trusted to a certain extent. Therefore, $TL_{initial}$ is assigned as the initial trust level, which would then increase or decrease according to user' behavior.

6.2.5 Impact Associated with Permissions

Every object in WBAC model is assigned with a security label (i.e., tagged as *protected* or *private* (Section 3.1.4)) which represents the level of object sensitivity. We assumed all *private* objects are categorized as highly sensitive. An object's level of sensitivity can be assigned by the object's owner (e.g., patient and/or healthcare providers). For example, consider clinical case study 1 (presented in Section 1.2) and patient *Jones*'s psychiatric notes. *Jones* could assign (or ask the psychiatric healthcare professional/institution) to assign his psychiatric notes the security level *highly sensitive*. This is also to comply with HIPAA principles which state that psychiatric notes are considered to be very sensitive information, which could have a higher security level compared to patient personal information (e.g., phone number and address) [307, 367]. Note that *protected* objects may also contain sensitive information (if the information is required for treatment). Moreover, the object sensitiveness is determined by the context and is influenced by individual preferences (vary from one individual to another), popularity (e.g., politicians, actors and actresses), social norms, etc. [212].

In this section, we calculate the impact of permissions on confidentiality, integrity, availability and compliance of an object based on the sensitivity level of the object and the likelihood of the threat occurrence. A permission is defined (Section 4.2.1) as a tuple (obj, opr) where obj is an resource in the organization and opr corresponds to an operation that a user can perform on the object. Objects are susceptible to different threats such as loss of integrity, loss of confidentiality, loss of availability and loss of compliance (Section 6.1.1). Intuitively, different objects have different security requirements that depend on the sensitivity level. For instance, some objects require that their integrity be well guarded, while other objects are confidential (their leakage would result in damage to the patient/healthcare providers). Hence, the risk exposure of the healthcare organization depends on the operation that is performed on the object and the level of sensitiveness. The risk value of an access request is the likelihood of threat occurrence multiplied by the impact if the permission is misused. We are interested in the residual risk, which means that the likelihood of a particular threat depends on the user trust level. Note that in most risk assessment processes, the likelihood of a threat depends on the mitigation mechanisms and controls that the organization has in place to reduce the vulnerabilities that can lead to the threat.

Definition 7: *$Threat = \{T_{confidentiality}, T_{integrity}, T_{availability}, T_{compliance}\}$ is a set containing sets of all possible threat events that can lead to misuse of an object through the operation and potentially cause damage to the patient/organization.*

Table 6.1: The impact of operations on different kinds of data

Operation	Object Sensitivity Level	Confidentiality	Integrity	Availability	compliance
Create	*Private/Protected*	0	1	1	1
Read	*Private*	1	0	0	1
Read	*Protected*	0	0	1	0
Write	*Private/Protected*	0	1	1	1
Delete	*Private/Protected*	0	1	1	1

$T_{confidentiality}$ is a set of threats to data confidentiality, $T_{integrity}$ is a set of threats to data integrity, $T_{availability}$ is a set of threats to data availability and $T_{compliance}$ is a set of threats to data compliance with law and regulations. We assume that the set of threats has all the possible threat events. In this case, it is also assumed that a threat modeling process is done to identify and prioritize all the threats based on vulnerabilities in the system and the expected impact on the organization's objects (patient's EHRs). For instance, in our work with Centre for eHealth, the most widely accepted threat modeling process, which has been proposed by Microsoft, is used to identify all possible threats to a telehealth trial system [2].

Definition 8: *The impact of an operation $Impact_{T,i}$ on the object is assigned the value 1 or 0, where 1 means that the operation has an impact on the object and 0 means that it does not.*

The model of *Sharma et al.* [327] (Table 6.1) is based on the impact of a requested operation based on confidentiality, integrity, and availability of an object. To instantiate this model, we adapted the table and used four impact quantification functions defined as (6.2), (6.3), (6.4), and (6.5). Let $q = (usr, obj, opr)$ (Section 4.4.1) where *usr* is user attribute, *obj* is object attribute, and *opr* is operation attribute, OBJ_A is a set of *private* objects and OBJ_B is a set of *protected* objects.

$$Impact_{T_{confidentiality},i}(q) = \begin{cases} 1 & \text{If } opr = Read \wedge obj \in OBJ_A \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

where $i \in T_{confidentiality}$.

$$Impact_{T_{integrity},i}(q) = \begin{cases} 0 & \text{If } opr = Read \\ 1 & \text{otherwise} \end{cases} \tag{6.3}$$

where $i \in T_{integrity}$.

$$Impact_{T_{availability},i}(q) = \begin{cases} 0 & \text{If } opr = Read \wedge obj \in OBJ_A \\ 1 & \text{otherwise} \end{cases} \tag{6.4}$$

where $i \in T_{availability}$.

$$Impact_{T_{compliance},i}(q) = \begin{cases} 0 & \text{If } opr = Read \wedge obj \in OBJ_B \\ 1 & \text{otherwise} \end{cases} \tag{6.5}$$

where $i \in T_{compliance}$.

According to Table 6.1, the impact of operation on availability, integrity, confidentiality and compliance of an object are assigned values of 1 and 0 based on data sensitivity levels with respect to different operations. Value 1 means the impact is high (required high level of protection) and 0 means low impact. Note that the impact is only considered from the security and privacy points-of-view, not from patient safety (i.e., a discipline that emphasizes safety in health care through the prevention and analysis of medical error) viewpoint. Patient safety has high priority (cf. report from *National Patient Safety Agency* for patient safety risk matrix [260]).

6.2.6 Risk Value Calculation

Risk is defined as a function of threat likelihood and impact. We determine the threat likelihood based on the user trust level (Section 6.2.4) and the impact of an operation on confidentiality, integrity, availability, and compliance as returned by the impact functions above. Given a user trust level and the impact of an operation, we calculate the risk associated with an access request as shown in 6.6:

$$Risk(q) = \left(\frac{\sum_{T \in Threat} \sum_{i \in T} (1 - TL(usr)) \cdot Impact_{T,i}(q)}{\sum_{T \in Threat} \mid T \mid} \right) \tag{6.6}$$

where T is all events in the set *Threat* (Definition 7), $1 - TL(usr)$ is the likelihood of a particular events occurrence (calculated based on trust $TL(usr)$ of user), $Impact_{T,i}(q)$ is the impact of an operation based on the threat T (Definition 8) and $q = (usr, obj, opr)$ (*usr* is user attribute, *obj* is object attribute, and *opr* is operation attribute). In 6.6, we subtract $TL(usr)$ by 1 because we assume that, if the trust level of user is high, then the likelihood of threat occurrence is low and vice versa.

6.2.7 Risk-Aware Access Decision Mechanism

After calculating the trust level of user and the risk value, The risk parameter is added to the decision function (4.11) shown in Section 4.4.4. It is said that a threat exists if a $usr \in USR$ can access an $obj \in OBJ$ such that $Risk(q) \geq risk\text{-}appetite(obj, opr)$.

Definition 9: *Risk-aware access decision mechanism is defined as if the access request would be granted based on the policy and the risk value $Risk(q)$ is less than $risk\text{-}tolerance(obj, opr)$, the access request is permitted, otherwise, denied. Formally (where Γ, Q and PolicySet are defined in chapter 4):*

$$rdf : \Gamma \times Q \times PolicySet \times Risk(q) \rightarrow Decision$$

$$rdf(\gamma, q, PolicySet, Risk(q)) = \begin{cases} \text{Permit} & \text{if } Case1 \\ \text{Mitigated} & \text{if } Case2 \\ \text{Deny} & \text{if } Case3 \\ \text{Indeterminate} & Otherwise \end{cases} \tag{6.7}$$

where,

- Case 1: $usr, obj, opr \in \gamma \mid \exists q \in Q, \exists policyset_{Id} \in PolicySet : Risk(q) \leq risk\text{-}appetite(obj, opr) \wedge f\text{-}WBAC(\gamma, policyset_{Id}, q) = permit$.
- Case 2: $usr, obj, opr \in \gamma \mid \exists q \in Q, \exists policyset_{Id} \in PolicySet : risk\text{-}appetite (obj, opr) \leq Risk(q) \leq risk\text{-}tolerance(obj, opr) \wedge f\text{-}WBAC(\gamma, policyset_{Id}, q) = permit$.
- Case 3: $usr, obj, opr \in \gamma \mid \exists q \in Q, \exists policyset_{Id} \in PolicySet : Risk(q) > risk\text{-}tolerance(obj, opr) \vee f\text{-}WBAC(\gamma, policyset_{Id}, q) = deny$.

That is, if the risk value given by $Risk(q)$ falls between the first and second bands (Figure 6.2) and there exists rules in the access policy (Section 4.4.1) to permit the request, then user *usr* is permitted to perform operation *opr* on object *obj* with risk value $Risk(q)$ and risk mitigation plans, otherwise, the access request is denied unless the request is approved by the object owner or the system administrator. Note that, as we mentioned earlier, in our experiments we have implemented the rules necessary for permitting access when a request is matched, PDP response notApplicable/indeterminate is interpreted as a deny response as PDP is configured as deny-based.

6.3 Risk-Aware Model Evaluation

6.3.1 Analysis

In this section, we analyze the security of risk-based decision methods against threats of allowing illegitimate accesses (Definition 2) and restricting legitimate accesses. We consider access state $\gamma \in \Gamma$ (Definition 1) that contains all the information necessary to make access control decisions for a given request and modify Definition 2 as follows:

Definition 10: *Access by user to object is considered to be legitimate if:*

(i) User and permission are included in the access state (Definition 1), and

(ii) Policy does permit the access, and

(iii) $Risk(q) \leq$ *risk-tolerance*(obj, opr).

Healthcare providers having legitimate access can be broadly categorized into two types [30, 378]: (1) honest healthcare providers and (2) malicious healthcare providers, based on the history. Honest healthcare providers intend to access patients health records that are *relevant and necessary* to fulfill their task in the patients' treatment. On the other hand, malicious healthcare providers do what the honest healthcare providers do except that he/she sometimes intentionally accessed the patients health records that are irrelevant to their tasks (an example shown in Figure 2.4). In our model, an authorized user is considered to be honest if $\mid RH(usr) \mid > \mid PH(usr) \mid$. On the other hand, an authorized user is considered to be malicious (not behaving according to rule) if $\mid PH(usr) \mid > \mid RH(usr) \mid$.

We simulated the user trust level based on the user history for the two types of healthcare providers (honest and malicious healthcare provider). We than compare the average of scores of the two types. For our experiments, since we do not have real-health history records, we assigned a user (*Dean*, *Bob*, *Cara*, *Alex*) random history values (100 events each) in their event history sets.

Example 6.1 (**User permission with risk assessment**)**:** *Let assume an example where Cara requests read access to file in* obj_b *and let*

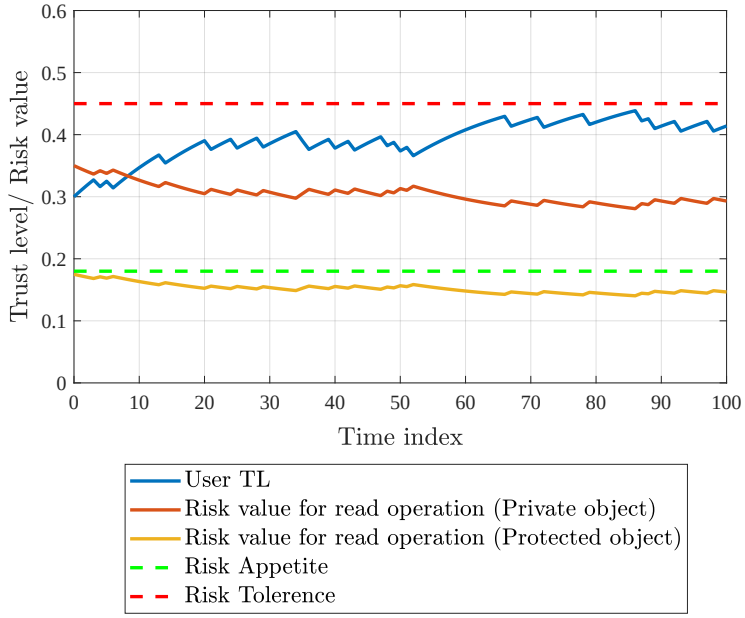
$$\begin{aligned} q = \{&(\textit{subject-id}, \textit{Cara}),\\ &(\textit{subject-TeamRole}, tr_t),\\ &(\textit{resource-type}, \textit{AliceProtected}),\\ &(\textit{action-id}, \textit{read})\}. \end{aligned}$$

Access state γ_1 (Example 4.5) shows that *Cara* is a member of team t_1 and she is assigned to team role tr_t. According to γ_1, we could say that the system does not violate the any constraints (e.g., SoD and cardinality constraints), and based on access control algorithm 2, *Cara* could access and preform the read operation on *AliceProtected* as she is a member of a team, she hold a team role and the policy allows the access (according to policy in Listing 3.4).

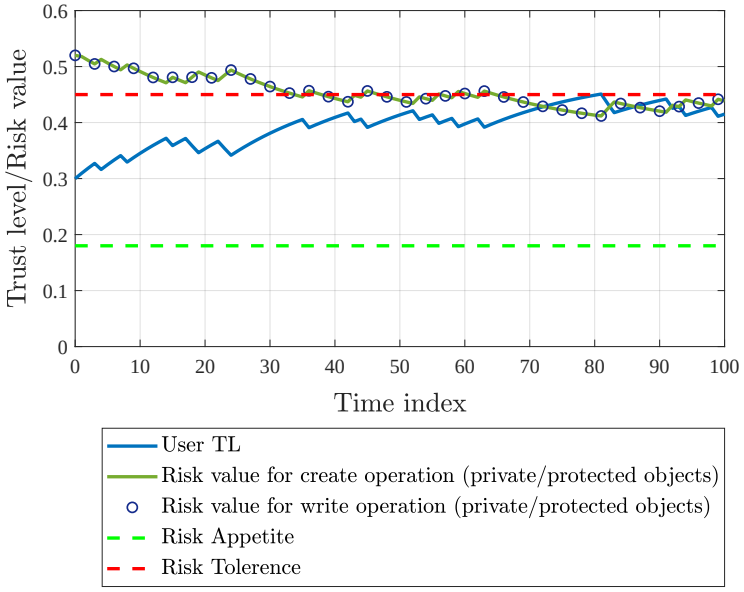
Considering the user trust level as the basic criterion for conducting risk assessment, we could see how trust level of user and risk values increases and/or decreases with the change in the user behavior (Figure 6.3). In Figure 6.3, we assumed that *Cara* is 80% behaving according to rules (honest healthcare provider) and she only violates the system rules and policies about 20% (i.e., 20% misbehaving user). Figure 6.3a indicates that the risk value $Risk(Cara, AliceProtected, read)$ is less than $risk\text{-}appetite(obj, opr)$ in case of the read operation on protected and private objects (Table 6.1). Therefore, for *Cara*'s request, it can be concluded that permitting *Cara* to read *AliceProtected* has low risk (Figure 6.3a) comparing with her trust level, which was calculated according to her history of rewards and penalty points. Moreover, as shown in Figure 6.3a, *Cara* did not pose a high risk when reading *private* objects where $Risk(Cara, AliceProtected, read) \leq risk\text{-}tolerance(obj, opr)$. Nonetheless, *Cara* should not have access to *private* objects (unless needed for the treatment) based on her team role *thought*. Therefore, we could say, in the present of WBAC access states (Definition 1) and the risk-aware model, illegitimate access (unauthorized access and improper access) is not possible.

On the one hand, in figure 6.3b, we considering the risk of *Cara* creating and writing to *private* and *protected* objects (Table 6.1). As Figure 6.3b illustrates, the risk of *Cara* writing and creating of objects is higher than the risk tolerance. This is due to the high impact of integrity, availability and compliance threats of the operations (Table 6.1). However, the risk decreases as the trust level of *Cara* increases. In this case, we assume *Cara* is a 20% misbehaving healthcare provider. In the best case, when a user does not have any penalties, then she will get the maximum trust value. Since there are no penalties history, the value of penalties becomes 0 in 6.1. Therefore, we get $TL(Cara) \approx 1$ (in case $\lambda_{RE} = 0.95$) and the $Risk(Cara, obj, opr) \approx 0$. Again, we say that in the present of WBAC access states (Definition 1) and according to *Cara*' team role, she will not be able to perform write and create operations on both *private* and *protected* objects. Therefore, illegitimate access (unauthorized access and improper access) is not possible.

Figure 6.4 shows the case when *Cara* is 80% misbehaving user (malicious healthcare provider). As shown in the Figure 6.3b, *Cara* is posing a risk towards



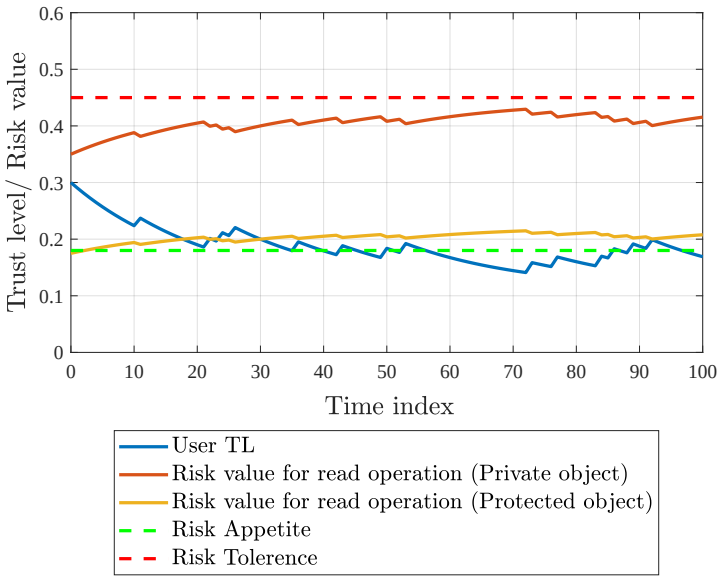
(a) Risk value in case of read operation



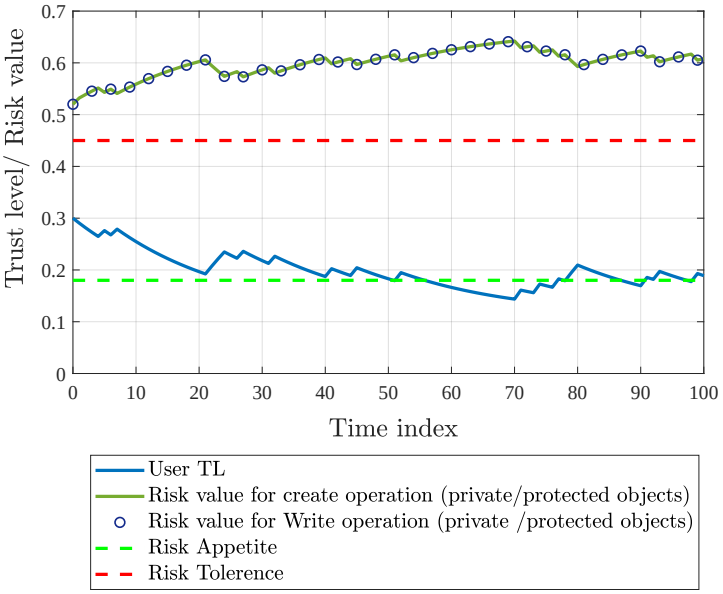
(b) Risk value in case of create and write operations

Figure 6.3: Trust level and risk value in case of 20% misbehaving user

protected and *private* objects since $TL(Cara)$ is low. However, according to access state γ (Definition 1), *Cara* is not permitted to access *private* object based on her team role. Therefore, in this case, unauthorized access is not possible. Moreover, since *Cara* is permitted to perform read operations on *protected* objects based on the access state γ and policy (Table 3.1), *Cara* will be permitted with risk mitigation plans (Section 6.1.1). The risk is mitigated by: (1) the patient *Alice*'s permission (i.e., *Alice* knows that *Cara* is a member of her treatment team and we assume that



(a) Risk value in case of read operation



(b) Risk value in case of create and write operations

Figure 6.4: Trust level and risk value in case of 80% misbehaving user

Alice has given a personal permission for *Cara* to access her protected objects), (2) WBAC model allows *Cara* to access restricted information on *Alice*'s protected objects, and (3) upon conducting risk assessment of *Alice*'s objects, it is noted that *Cara* poses a threat to this *protected* objects. Therefore, the problem of low data breaches detectability is examined and *Alice* or the system administrator has discovered the purpose of the access is for *Alice*'s treatment.

Figure 6.4b represents a situation where *Cara* is posing a high risk towards *protected* and *private* objects in case of creating and writing operations. As showed in

Table 6.2: Comparison summary of different risk-aware access control models

Risk model	**Risk factors**				
	User trust level	Object security level	Impact of operation	Past user behaviors	Risk boundaries
Sandhu [316]	Yes	Yes	No	No	No
Cheng, et al. [85]	Yes	Yes	No	No	Yes
Ni, et al. [263]	Yes	Yes	No	No	Yes
Li, et al. [224]	No	Yes	Yes	No	No
Ma [233]	Yes	Yes	Yes	No	No
Bijon, et al. [53]	Yes	Yes	No	No	Yes
Baracaldo and Joshi [37]	Yes	No	Yes	Yes	Yes
Shaikh et al. [326]	Yes	Yes	No	Yes	No
Risk-Aware WBAC	Yes	Yes	Yes	Yes	Yes

the Figure 6.4b, the risk is greater than the risk tolerance. In this case, *Care* is denied to perform create/write action on both *protected* and *private* objects. In the worst case, when the user does not have any rewards points, the user will get the minimum trust value TL_{min}. According to 6.6, the likelihood of threats would be high and thus, the risk value $Risk(q)$ would be greater than the $\textit{risk-tolerance}(opr, obj)$. In function 6.7, if $Risk(q) \geq \textit{risk-tolerance}(opr, obj)$ the access would be denied unless approved by the object owner or security administrator. Therefore, we could say that illegitimate access is not possible. However, as we mentioned early (section 6.2.2), we can not prevent a healthcare provider from accessing an object during patient treatment. In this case, in our access control model, the risk that *Cara* poses to the object has been mitigated by the fact that (1) the current WBAC access state γ does not allow *Cara* to create/write to any object because operation create/write by *Cara* is not needed for the treatment and that (2) in case *Cara* wants to create/write to *Alice*'s objects, the request would be investigated by the team coordinator and, if needed, *Cara* would be get access with a personal permission.

6.3.2 Comparison with Related Work

As discussed in Section 6.1.2, there are numerous research directions on risk-aware access control models. In this section, the proposed risk-aware WBAC model is compared with these models to better understand the differences between them. Table 6.2 summarizes the discussion and presents the comparison results. It contains the risk model and the risk factors used to estimate the risk value. The table uses *Yes* and *No* to indicate whether the model facilitates the concerned risk factors. As shown in the Table 6.1 and according to our discussion in Section 6.1.2, fuzzy logic models [85, 224, 263] consider all subject-object accesses to include the temptation to leak information and aim to quantify the risk of unauthorized disclosure of in-

formation by subjects. They are tolerant of imprecisely defined data. In healthcare systems, some patient information, such as blood test results, might be confidential, whereas another part, such as geographical information, might be unclassified. In comparison with these models, the aim of our model is to assess the threat posed by subjects towards objects by referring to the user trust level, object sensitivity and the impact of the operation on the object. Moreover, the presented fuzzy models [85, 224, 263] and many others cannot cope with the changing behavior of the users to detect insiders activities due to their limited consideration of risk factors. Our model differs from these work by considering the risk factors and trust levels of users depending on their behavior. Furthermore, WBAC model support the *minimum necessary* and *need-to-know* principles and was extended with risk-aware model to assess threats in subject-object accesses.

6.4 Chapter Summary

The motivation behind creating a risk assessment framework for WBAC is to help enhance the system security in terms of protecting healthcare information from insider threats, such as patient data disclosure and unauthorized access or modification by insiders. The main goal of our risk assessment framework is to evaluate the risks associated with access requests and effectively enforce the principles of *need-to-know* and *minimum necessary* in a collaborative healthcare system. We assert that risk assessment does not prevent security policy violations but can detect violations and help determine the type of corrective action needed. To conclude, our model was able to detect the little difference in user changing behavior which gives the object owners the ability to detect any malicious events.

Chapter 7

Conclusions and Future Work

This chapter summarizes the main ideas and the findings of our research. Also, it presents possible future directions followed by conclusions of this thesis.

7.1 Discussion and Observations

MDTs are likely to benefit everyone, but for such teams to keep working well, skills and sufficient coordination as well as resource management are needed. Evidence showed that EHRs may improve the work within MDTs, through which healthcare providers share healthcare information more easily and work together as a team to solve particular medical cases. However, the EHRs might also leave patients more susceptible to privacy violation where confidential information is improperly accessed and exploited by MDT members. The main challenges around the sharing of sensitive patient data is (1) the assignments and revocation of permissions and access rights to healthcare providers, especially to outsiders (healthcare providers from different healthcare organization), (2) permissions and access rights should also restrict access to only the portion(s) of data intended for the patient's treatment, since unauthorized disclosure or improper access of highly confidential health data can be devastating and is against many regulations. Evidence also showed that it is challenging to predefine all access needs for MDTs based on the subject-object model. One example of such a situation is explained in our scenarios, which may not be predictable and it would be hard to express the condition of who should join the MDT. Moreover, in deciding on the extent and limit of health information sharing, for instance, in the case of patients *Jones*'s and *Alice*'s treatment, which sensitive information should be disclosed to an assisting healthcare provider so that

collaboration can be effective, and which should be hidden to safeguard the patient's privacy?

This thesis addresses access control matters in collaborative engagements with complex scenarios in the collaborative healthcare domain. The focus is mainly on collaborative activities that are accomplished by organized groups of healthcare providers (MDTs) within or among healthcare organizations with the objective of accomplishing a specific work (a case of patient treatment). The main goal is to provide an access control model that enable a balance between collaboration and safeguarding sensitive patient information. In the following subsection, critical observations of this study are discussed. Moreover, research questions raised in Chapter 1 are answered one by one based on the research objectives reached in the subsequent chapters.

7.1.1 Observations

There are certain observations that we have learned from the previous studies that was highly considered during this research. Observations as follows:

1. **What do patients and healthcare providers want from EHRs?** From the patient perspective, patients found that EHRs are useful and acceptable. The majority were concerned about security and confidentiality, including access and disclosure of their records. It is clear that, on the one hand, patients want EHR systems to make health data accessible, available and easy for healthcare provider to find and use. However, they also want to be informed regarding access, disclosure and use of their data. From the perspective of healthcare providers, they want EHRs to make their practice work better, easy to manage and be able to coordinate patient care easily by communicating with one another, deciding who will be doing what interventions and then sharing the information in the most effective way that EHRs facilitate.

2. **Consent as a basis for treatment: When can it be used and when should it not be used?** As we explained in Section 2.3, patient consent is required by many standards and by legislation. Moreover, there are many requirements for patient consent to be legal. First, a valid consent must be given implicitly or explicitly (Requirement 1) and must be no advantages or disadvantages associated with the consent. If a patient denies to give a consent to access a specific information, there must be no negative consequences for the patient other than that the healthcare provider fails to access the required information. Second, the consent must be specific for the purpose of the treatment and

separate consents must be obtained for different purposes. But, in healthcare environments, especially when a patient is in life-threatening illness or injury (e.g., patient is unconscious), explicit consent is difficult to obtain. Either way, managing patient consent during patient treatment is already complex. It adds an extra level of complexity to all access control models. This is due to legal requirements of valid consent. First, under the terms of HIPAA [6], valid consent to use or disclose health information must contain "a description of the information to be used or disclosed"; "the name of the person or entity authorized to make the use or disclosure"; "the name of the person or entity to whom the disclosure may be made"; "a description of each purpose of the requested use or disclosure"; "an expiration date or expiration event" and "the signature of the individual and date". Second, discrepancy is possible between the set of information that patient consents to share and the set of information that healthcare providers have access to. Finally, as discussed earlier, the main requirement for protecting privacy in HL7 and others [333] is that health data sharing must be controlled by patient consent while allowing differential access to aspect of health information depending on the sensitivity of the information as perceived by the patient.

3. **EHRs require better ways to securely exchange information:** EHRs are promising to be an ideal solution for addressing the information exchange challenges that today's MDTs are facing. It provides an automated and fast information exchange between healthcare providers within or among healthcare organizations. However, security and privacy mechanisms to ensure secure interoperable EHR applications are slowly beginning to emerge. For access (uses and disclosures) of patient health information, access control policies and procedures must be in place to identify and authorize a healthcare provider or MDT member who needs access to the health information to carry out their job duties. Also, it is necessary to identify and authorize the types of information needed and conditions appropriate to their access. For example, access control policies should permit only doctors or others involved in the treatment to have access to patient medical records, based on *need-to-know* and *minimum necessary* principles.

4. **What is good for security is not necessarily useful for MDT practice?:** Bridging the gap between security requirements and MDT practice is a critical focus for security researchers. This is a challenge because what is good for security is not always what healthcare providers want. On the one hand, healthcare provider (members of MDTs) need tools such as EHRs to provide

an easy sharing of health information, real-time access to health records and easy to use. On the other hand, security seeks to ensure the health records' availability, confidentiality, and integrity while providing them only to those with proper access rights. Security researchers, specifically in access control and authorization, have made the best effort to propose an access control model that balances between security and MDT requirements. Yet, these models do not always meet the needs of MDTs due to the inconsistencies that exist within the MDT workflow and these models' approaches.

As a result of these observations, it could be concluded that, if we do not coordinate the MDT and shared information, we cannot coordinate the patient care, and if we do not coordinate the patient care, we will have inefficiency and poor healthcare quality.

7.1.2 Answers to Research Questions

Answer 1: *to RQ 1 (What health information (patient EHRs) should be available and under what circumstances can health information be shared during MDT collaboration?) and RQ 2 (Who should decide on the extent and limits of health information sharing?).*

To develop a new access control model for supporting the MDTs members who are involved in a particular patient treatment, the first step that needs to be done is analyzing the domain (MDTs work in healthcare) where an access control model will be built and applied. Here, domain analysis aims at extracting, identifying, capturing, organizing and making reusable information about the domain. Therefore, in this thesis, analysis of the MDTs work with a respect to the access control model was done to gain the deep knowledge of the domain. It is understood that one of the key aspects of an MDT is sharing of patient health information. To cooperate, each MDT member must be prepared to receive, gather and share their findings with the other team members based on the patient's case. To answer the question *What health information (patient EHRs) should be available and under what circumstances can health information be shared during MDT collaboration*, we presented two clinical case scenarios. This is because a general answer does not exist for the posed question due to the diversity of the healthcare domain and MDTs' work. From the two clinical case scenarios and other studies in the literature, we understand that patient information needs to be available to healthcare providers during the treatment course (treatment pathway explained in Table 2.1). Yet, *what is relevant* and *when is it relevant* are ambiguous concepts that are highly dependent

on different the patient cases and, therefore, hard to predefine. As we discussed in Section 2.2.2, the relevancy and the necessity of any information depends on the frequency of healthcare providers exposure to the problem being addressed and the type of evidence presented.

Speaking of information availability and to answer question 2, it was noted that, the obligation to protect patient privacy is forced by legislative institutions of most countries and it is a duty of healthcare providers. According to good medical practices standards and legislation (HL7, HIPAA, the Norwegian code of conduct for information security, the UK Good Medical Practice etc.), the patient should not be identified to greater extent than is necessary to fulfill the purpose of the treatment. In this case, therefore, patient health information and identifying information should be kept separate. Moreover, the most sensitive health information should be shared and disclosed based on the patient's consent (see HIPAA Privacy Rules [307, 366, 367], Personal Health Data Filing System Act [387] and the Norwegian code of conduct for information security [265]). Also, according to such standards and legislation, we understand that it is important to find a good balance between how health information is processed (shared and disclosed etc.) and the patient's privacy be maintained. For example, the report entitled "*Access to health records: Guidance for health professionals in the UK*" [65] indicates what information should not be disclosed. Thus, we could argue (based on standards, legislation and presented clinical case scenarios) that health information needs to be available during the patient treatment. However, not all of the health records should be available. The availability of health information can change during the patient pathway (Table 2.1) and every healthcare provider (member of MDTs) can decide on the extent and limits of health information sharing (more details in Section 2.2.2).

Speaking of who should decide on what information is necessary and relevant, according to legislation (cf. study on the legal framework for interoperable eHealth in Europe [107]), decisions regarding the right of access (*necessary and relevant*) must be taken by a policy authority and the healthcare providers (organizations or healthcare professionals) who possess the patient records. Moreover, as described in Section 2.2 (Table 2.1), the *case manager* in the case management model should decide and assign the case (treatment of the patient) to appropriate healthcare providers. Also, the *case manager* can decide on the *necessary and relevant* of health records for the case. In the case of key worker models, the *team leader* or *team manager* decides. Note that other team members can also decide if they think certain records (health information) are necessary and needed for the case. However, this should be done through the *team manager* or *team coordinator*.

Answer 2: *to RQ 3 (What are the strengths and weaknesses of existing access control models proposed for healthcare?).* As we have discussed earlier, access control is ideal for managing access to information and controlling the activities of legitimate users to access a resource in the system. The ultimate goal of an access control system is to allocate all users the specific access level necessary to do their job. Access control mechanisms have undergone many developments in both academia and industry in order to meet collaborative system needs. The models include MAC, DAC, RBAC, ABAC and many others. DAC defines access control privileges based on the subject's identity and the access rules in place. It determines whether the subject can or cannot execute particular actions on specific resources. DAC allows the subject to own resources and permits ownership transfer to another subject. Although DAC policies tend to be flexible and are widely deployed, DAC has several drawbacks when utilized in collaborative healthcare systems. First, ownership and permission updating is not scalable as the number of users (e.g., healthcare providers and patients) and resources (e.g., health records) are increases continuously. Second, DAC policies do not provide high security assurance because granting read access is transitive and DAC allows data to be copied from one resource to another, which can result in unintentional information flow in the system. Unlike in DAC, in MAC, access rights to objects are decided upon by a central authority. MAC controls the information flow to ensure information confidentiality and integrity. However, enforcing MAC policies in collaborative healthcare systems is often difficult due to the vast numbers of users and wide range of resource types.

The motivation for RBAC is to address the perceived deficiencies in existing discretionary and mandatory access control models in terms of specification and enforcement of organization-specific access policies as well as reducing the complexity and cost of administering systems based on these models. The two main advantages of RBAC are the simplification of privilege management and the presentation of a high level view of security in an organization. Although the RBAC model has several advantages, it also has disadvantages in attempting to apply RBAC in collaborative healthcare environments (discussed in Section 1.1.4). In short, RBAC does not seem to have enough power to express the wide range of security requirements and to capture fine access control granularity in EHRs and MDTs environments. Therefore, RBAC has been extended to support diverse domains in data authorization management with various constraints. The extensions include task-role based, team-based, contextual-role based, context-aware and others (discussed in Section 2.5 and 2.6). However, these extended models incur additional complexity to collaborative systems because they still face some problems.

ABAC uses subject attributes (e.g., name, ID, and role in organization) and object attributes (e.g., metadata properties) to provide authorization. ABAC overcomes the user role assignment problem existing in RBAC and focuses on the attributes of a user requesting access. It is a flexible model that is considerably easier to administer than RBAC. However, the greater flexibility comes with higher complexity due to the specification and maintenance of the access policies. Granularity and manageability are inversely proportional to one another. Higher granularity in security invariably implies more complex management. This is apparent in ABAC, which offers higher control or granularity at the expense of lower manageability. On the other hand, RBAC evidently provides lower granularity for better manageability. As showed earlier in this thesis, in collaborative environments such as healthcare, it is not easy for traditional authorization mechanisms like RBAC and ABAC alone to specify authorization constraints due to the complexity of a continuously growing number of users, health records, lack of granularity and manageability as well as requirements for flexibility in the specification and maintenance of policies. As authorization policies are becoming less manageable and more complex, the possibility of information leaks caused by improperly designed authorization policies increases. Thus, additional authorization constraints must be added to traditional access control models to prevent some of the information leaks caused by these policies. Moreover, it is important that an access control model should ensure shared information confidentiality and also avoid adding administration and management complexity. These features can be accounted for by extending RBAC and ABAC to WBAC, in order to support MDTs' work, as shown earlier in this thesis.

Answer 3: *to RQ 4 (How can the access control model be extended to support MDT collaboration and health information sharing without adding administrative overhead?).* The WBAC model is proposed by introducing the team role concept and modifying the user role assignment model from RBAC and ABAC models. In WBAC, the notion of team role is used to solved the problem of inter-organization definitions of RBAC roles and profiles that do not exist in the internal organization (problem discussed in Section 3.1.2). For RBAC, healthcare organizations must create and agree on a collaborative roles and role profile, which define all the necessary permissions the user (healthcare provider) will get once he/she is assigned the role. In WBAC, team role classification based on *Belbin*'s team role theory would help healthcare organizations on the role definitions and role profiling. For example, considering the problem discussed in Figure 1.2, if the specialized hospital wants to collaborate with GPs, the specialized hospital does not have to define GPs roles and profiles. Our proposed team role would ensure finer roles of collaboration. The

team role of each team member will subsequently determine the extent of access a member may receive. The level of fine-grained control of access (granularity) to objects that can be authorized to healthcare providers is managed and controlled based on the work required.

WBAC can cater for special security requirements such as the support of private access rights for each of the team members in the same team, and the differentiation between the access rights of healthcare providers is associated within the team role. We have considered possible means (such as following the RBAC standards and medical guidelines etc.) to make this an appropriate trade-off that will retain the benefits of RBAC and ABAC while extending their ability to support MDTs' work. WBAC meets our expectation of allowing fine-grained access control as well as enhances the practicability and manageability of access control in MDTs environments. That is, performance evaluation showed that WBAC is performing as well as ABAC and RBAC in this area.

7.2 Evaluation Against Insider Threats

Insider threats are categorized according to the following types: unauthorized access threats and improper access threats. Here, we evaluate our access control model against these two types of insider threats.

7.2.1 Unauthorized Access Threats

For unauthorized access threats, insiders would (or try to) access healthcare data to which they have no authorized access. According to our threat modeling in [2], this could happen by obtaining credentials from authorized users, for example stealing credentials or devices that contain credentials of other healthcare providers. Healthcare providers sharing their login credentials with friends, relatives or other healthcare providers may also have potential impact, like credential misuse, tampering with patient data, or private information disclosure. In such cases, the insider is able to pass the authentication mechanism used in the system. We strongly believe that the robustness of access control models such as RBAC, ABAC and WBAC is dependent on the authentication mechanism (authentication is out of the scope of this thesis). However, in our studies [4, 5] (collaboration with another PhD project [399]), we proposed an authentication schema based on the proposed team role that is suitable for collaborative healthcare systems to address the issue of authenticated access. The proposed schema is based on attribute-based authentication

(ABA), which is a means of authenticating users by attributes or their properties. In this proposed scheme, *Dean*, described in our usage scenario, will first obtain a key from a trusted authority based on his attributes. Then, each team member should obtain their attribute keys from *Dean*. All these attribute keys are only active for the duration of a specific work (*Alice*'s treatment). When this work is finished, all attribute keys of the users in this group should be revoked. When the team members want to access documents (*Alice*'s objects), they generate a signature based on the required attributes defined in the access policy. If their signatures are valid, they satisfy the access policy and will be granted the required access.

Users outside the treatment team cannot generate a valid signature in our proposed scheme, therefore, users without the required attributes cannot generate a valid signature for successful authentication. As a result, healthcare providers who join the treatment team (e.g., *Cara* and *Alex*) should register themselves to obtain their authorization keys from *Dean*. Therefore, all healthcare providers who join *Alice*'s treatment will be identified by the team manager (*Dean*) and will be authorized access to *Alice*'s EHRs upon obtaining their authorization keys. In this case, WBAC reduces the risk of unauthorized access to EHR documents and confidentiality can be satisfied.

7.2.2 Improper Access Threats

Improper access threats include elevation of privilege, data tampering and disclosure of confidential data. With elevated privilege threats, insiders may attempt to elevate their privileges in order to gain additional access to the system components. For example, healthcare providers may impersonate the context of administrators in order to gain additional privileges and more control over the system. Data tampering refers to intentionally or accidentally modifying, adding and/or deleting data by insiders with over-privileges. Confidential data disclosure potentially occurs if sensitive data, such as patient health records, can be viewed by unauthorized users due to improper data protection.

In this thesis, our main focus was on over-privilege access because it is a common occurrence in today's healthcare environments and also represents a significant source of insider threats for healthcare organizations. For example, consider our usage scenario (Section 2.2.1) where *Cara* may not have access to the intended health information because maybe she cannot access due to access policy restriction. In this case, *Cara* is under-privileged and she may not be able to perform her task on *Alice*'s treatment. On the other hand, *Cara* may also be over-privileged and have access to unnecessary information. In this case, *Cara* will have more access than

what is really required and she might violate the privacy of patient Alice's.

In WBAC, access to health records is controlled via the roles, team roles and collaboration policies. Due to the fine granularity and flexibility of WBAC, the set of classified objects that healthcare providers can access is constrained by the team roles to which the healthcare providers are assigned and by the access policies. For instance, a collaboration policy was implemented (Listing 3.4) to permit *Cara*, who is assigned the *thought* team role, to read *protected* resources. Authorization policy authors must ensure that the policies are implemented correctly to safeguard the organization's objectives, because incorrect policy implementation will result in subjects having improper access privileges. With WBAC, it is feasible to specify that healthcare providers who have already joined a team and have been assigned team roles can access (e.g., read only) objects in the patient EHRs to which the healthcare providers were invited for the specific work (e.g., *Alice*'s treatment).

7.3 Limitations and Future Work

This section presents WBAC limitation and proposals for future enhancements.

7.3.1 Limitations of WBAC

It seems there is a tradeoff between simplifying the management of access rights and providing fine granularity; and between providing a higher level of security, and satisfy the requirements of easy access for MDTs. Providing a balance between these advantages in a WBAC model is greatly affected by many factors:

1. Object classification: Healthcare records contain a wide range of information and healthcare record classification is expensive as it requires skilled and knowledgeable people. Failure to achieve appropriate classification of healthcare records would render WBAC less secure (future work on this problem is suggested in Section 7.4).

2. Collaboration policy conflicts between collaborative parties: Collaboration pattern for information sharing is required to provide the set of rules regarding how the collaboration should be carried out. Guidelines and standards are also required to secure collaboration between the collaborative parties.

7.4 Future Work

This research work can continue in the following directions.

1. **Prototype the administrative operations to be implemented**: In this thesis, administrative operations [128] are not considered during the prototyping of the model. So, as future work, administrative operations will be developed and prototyped to understand the possible difficulties in managing the model during an actual implementation. The specification of the administrative operations would be based on NIST model of RBAC [128] as follows:

   (a) **Permission modification**: The process of modifying the sets of permissions associated with roles and team roles to meet the authorization requirements of an organization.

   (b) **Privilege modification**: The process of changing and modifying user membership (assignment relations) in a role or team role.

   (c) **Revocation process**: The process of revoking user membership in a role or team role. In WBAC, we also have team revocation, which deprives team of the work, user revocation which is a process to revoke a user from a team, and work revocation to map an active work to inactive.

   (d) **Permission review**: The process of reviewing all available permissions (role and team role) assigned to a particular user.

2. **Integration with resource classification models**: Machine learning approaches have become a major field of research in order to handle complex problems such as health information classification. As future work, we plan also to explore the feasibility of machine learning and data-mining approaches for the classification and resource allocation to enhance the predication of health information needs. Machine learning algorithm (e.g., neural network [391]) will be used to quickly determine the most likely health information are *relevant and necessary* for a specific patient case.

3. **Cross-Border Healthcare Collaboration:** cloud computing and information technology adaptation to healthcare is becoming increasingly important in many countries [99, 219]. European Union (EU) countries are seeking new ways to modernize and transform their healthcare systems by using EHRs in order to provide EU citizens (patients) with safe and high quality treatment in any EU country [70, 390] (EU directive 2011/24/EU framework on cross-border health care collaboration in the EU [116, 117, 118, 278]). Access to cross-border healthcare in the EU has undergone many developments in both academia and industry to meet EU healthcare domain needs. The eHealth Action Plan 2012-2020 [115] and the EU-funded project "UNIversal solutions in

TELemedicine deployment for European health care" (United4health) [365] are among such developments. The aim of these projects is to provide solutions to improve healthcare quality, provide access to a high-quality healthcare system to all EU citizens around Europe, and support close cooperation between healthcare professionals and care providers from different organization. Therefore, in future, the WBAC model can be further investigated in terms of cross-border healthcare collaboration. The plan is to evaluate the validity of WBAC to support secure cooperation between healthcare professionals and care providers from different organization at EU.

7.5 Conclusions

It is evident that EHRs have a great potential to support MDTs' work, including but certainly not limited to creating, managing and sharing patient healthcare information as well as facilitating an easy coordination and communication between healthcare providers, thus improving patient satisfaction and engagement. However, unauthorized disclosure and improper access to patient healthcare records are a major concern of this thesis when sensitive healthcare data is shared among a group of healthcare professionals within or across healthcare organizations.

WBAC was proposed to address these concerns and support the security and MDT requirements on access control. The major contributions of the WBAC model include ensuring that access rights are adapted to the actual needs of the healthcare providers and providing fine-grained control of access with the minimum necessary standard, whereby healthcare providers are granted minimal access to carry out their duties.

Although many challenges remain, WBAC seems to be a promising candidate indeed. We are optimistic that WBAC can handle dynamic environments and scales well. This conclusion is based on case evaluations. While this allows for optimism about the suitability of WBAC for use within collaborative healthcare environments, final judgment must be reserved until field tests have been conducted.

REFERENCES

[1] Aanestad, M., Grisot, M., Hanseth, O., and Vassilakopoulou, P. (2017). *Information Infrastructures within European Health Care: Working with the Installed Base*. Springer.

[2] Abomhara, M., Gerdes, M., and Køien, G. M. (2015). A stride-based threat model for telehealth systems. *Norsk informasjonssikkerhetskonferanse (NISK)*, 8(1):82–96.

[3] Abomhara, M. and Køien, G. M. (2015). Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security*, 4:65–88.

[4] Abomhara, M. and Yang, H. (2016a). Attribute-based authenticated access for secure sharing of healthcare records in collaborative environments. In *Proceedings of the Eighth International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED 2016)*, pages 138–144.

[5] Abomhara, M. and Yang, H. (2016b). Collaborative and secure sharing of healthcare records using attribute-based authenticated access. *International Journal on Advances in Security Volume 9, Number 3 & 4*, pages 148–195.

[6] Agris, J. L. (2014). Extending the minimum necessary standard to uses and disclosures for treatment: Currents in contemporary bioethics. *The Journal of Law, Medicine & Ethics*, 42(2):263–267.

[7] Ahn, G.-J. and Sandhu, R. (2000). Role-based authorization constraints specification. *ACM Transactions on Information and System Security (TISSEC)*, 3(4):207–226.

[8] Ahn, G.-J. and Shin, M. E. (2001). Role-based authorization constraints specification using object constraint language. In *proceedings of the 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2001)*, pages 157–162. IEEE.

[9] Ajami, S. and Bagheri-Tadi, T. (2013). Barriers for adopting electronic health records (ehrs) by physicians. *Acta Informatica Medica*, 21(2):129.

[10] Al-Kahtani, M. A. and Sandhu, R. (2002). A model for attribute-based user-role assignment. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 353–362. IEEE.

[11] Alhaqbani, B. and Fidge, C. (2008). Access control requirements for processing electronic health records. In *Business Process Management Workshops*, pages 371–382. Springer.

[12] Almehmadi, A. and El-Khatib, K. (2015). On the possibility of insider threat prevention using intent-based access control (ibac). *IEEE Systems Journal*, PP(99):1–12.

[13] Alqahtani, A., Crowder, R., and Wills, G. (2017). Barriers to the adoption of ehr systems in the kingdom of saudi arabia: An exploratory study using a systematic literature review. *Journal of Health Informatics in Developing Countries*, 11(2).

[14] Alqatawna, J. F., Rissanen, E., and Sadighi, B. (2007). Overriding of access control in xacml. In *Policies for Distributed Systems and Networks, 2007. POLICY'07. Eighth IEEE International Workshop on*, pages 87–95. IEEE.

[15] Alshehri, S. (2014). *Toward Effective Access Control Using Attributes and Pseudoroles*. PhD thesis, Rochester Institute of Technology.

[16] Alshehri, S., Mishra, S., and Raj, R. (2013). Insider threat mitigation and access control in healthcare systems. *Rochester Institute of Technology RIT Scholar Works*.

[17] Alshehri, S. and Raj, R. K. (2013). Secure access control for health information sharing systems. In *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*, pages 277–286. IEEE.

[18] Altuwaijri, M. M. (2008). Electronic-health in saudi arabia. *Saudi medical journal*, 29(2):171–178.

[19] Aminpour, F., Sadoughi, F., and Ahamdi, M. (2014). Utilization of open source electronic health record around the world: A systematic review. *Journal of research in medical sciences: the official journal of Isfahan University of Medical Sciences*, 19(1):57.

[20] Andrews, L., Gajanayake, R., and Sahama, T. (2014). The australian general public's perceptions of having a personally controlled electronic health record (pcehr). *International journal of medical informatics*, 83(12):889–900.

[21] Ankem, K. (2005). Types of information needs among cancer patients: A systematic review. *Libres*, 15(2):1.

[22] Annas, G. J. (2003). Hipaa regulations–a new era of medical-record privacy? *The New England journal of medicine*, 348(15):1486–90.

[23] Appari, A. and Johnson, M. E. (2010). Information security and privacy in healthcare: current state of research. *International journal of Internet and enterprise management*, 6(4):279–314.

[24] Ardagna, C. A., Cremonini, M., Damiani, E., di Vimercati, S. D. C., and Samarati, P. (2006). Supporting location-based conditions in access control policies. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 212–222. ACM.

[25] Ardagna, C. A., Di Vimercati, S. D. C., Foresti, S., Grandison, T. W., Jajodia, S., and Samarati, P. (2010). Access control for smarter healthcare using policy spaces. *Computers & Security*, 29(8):848–858.

[26] Arka, I. H. and Chellappan, K. (2014). Collaborative compressed i-cloud medical image storage with decompress viewer. *Procedia Computer Science*, 42:114–121.

[27] Armando, A., Carbone, R., Chekole, E. G., and Ranise, S. (2014). Attribute based access control for apis in spring security. In *Proceedings of the 19th ACM symposium on Access control models and technologies*, pages 85–88. ACM.

[28] Arocha, J. F., Wang, D., and Patel, V. L. (2005). Identifying reasoning strategies in medical decision making: a methodological guide. *Journal of biomedical informatics*, 38(2):154–171.

[29] Asif, K., Ahamed, S. I., and Talukder, N. (2007). Avoiding privacy violation for resource sharing in ad hoc networks of pervasive computing environment. In *Proceedings of the 31st Annual International Computer Software and Applications Conference-Volume 02*, pages 269–274. IEEE Computer Society.

[30] Azaria, A., Richardson, A., Kraus, S., and Subrahmanian, V. (2014). Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data. *IEEE Transactions on Computational Social Systems*, 1(2):135–155.

[31] Babiker, A., El Husseini, M., Al Nemri, A., Al Frayh, A., Al Juryyan, N., Faki, M. O., Assiri, A., Al Saadi, M., Shaikh, F., and Al Zamil, F. (2014). Health care professional development: Working as a team to improve patient care. *Sudanese Journal of Paediatrics*, 14(2):9.

[32] Bacon, J., Moody, K., and Yao, W. (2002). A model of oasis role-based access control and its support for active security. *ACM Transactions on Information and System Security (TISSEC)*, 5(4):492–540.

[33] Bain, C. (2015). The implementation of the electronic medical records system in health care facilities. *Procedia Manufacturing*, 3:4629–4634.

[34] Balogh, E., Miller, B. T., and Ball, J. (2015). *Improving diagnosis in health care*. National Academies Press.

[35] Banerjee, A. and Naumann, D. A. (2004). History-based access control and secure information flow. In *Construction and Analysis of Safe, Secure, and Interoperable Smart Devices*, pages 27–48. Springer.

[36] Baracaldo, N. (2016). *Tackling insider threats using risk-and-trust aware access control approaches*. PhD thesis, University of Pittsburgh.

[37] Baracaldo, N. and Joshi, J. (2013). An adaptive risk management and access control framework to mitigate insider threats. *Computers & Security*, 39:237–254.

[38] Bardhan, I. R. and Thouin, M. F. (2013). Health information technology and its impact on the quality and cost of healthcare delivery. *Decision Support Systems*, 55(2):438–449.

[39] Barnes, S.-A., Green, A. E., Batty, E., and Pearson, S. (2017). Key worker models: what key worker approaches, capacity and capabilities are important at different stages of the journey to employment?

[40] Bath, P. A. (2008). Health informatics: current issues and challenges. *Journal of Information Science*, 34(4):501–518.

[41] Baus, A. (2004). Literature review: barriers to the successful implementation of healthcare information systems. *Office of Health Services Research, West Virginia University Department of Community Medicine, Morgantown, WV*.

[42] Belbin, R. M. (2012). *Team roles at work*. Routledge.

[43] Bell, D. E. and LaPadula, L. J. (1975). Computer security model: Unified exposition and multics interpretation. *MITRE Corp., Bedford, MA, Tech. Rep. ESD-TR-75-306, June*.

[44] Benaloh, J., Chase, M., Horvitz, E., and Lauter, K. (2009). Patient controlled encryption: ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 103–114. ACM.

[45] Bergen, A. (1992). Case management in community care: concepts, practices and implications for nursing. *Journal of advanced nursing*, 17(9):1106–1113.

[46] Berner, E. S. and Moss, J. (2005). Informatics challenges for the impending patient information explosion. *Journal of the American Medical Informatics Association*, 12(6):614–617.

[47] Bertolissi, C. and Fernández, M. (2010). Rewrite specifications of access control policies in distributed environments. In *Security and Trust Management*, pages 51–67. Springer.

[48] Bertolissi, C., Fernández, M., and Barker, S. (2007). Dynamic event-based access control as term rewriting. In *Data and Applications Security XXI*, pages 195–210. Springer.

[49] Bhartiya, S., Mehrotra, D., and Girdhar, A. (2015). Proposing hierarchy-similarity based access control framework: A multilevel electronic health record data sharing approach for interoperable environment. *Journal of King Saud University-Computer and Information Sciences*.

[50] Bhatti, R., Bertino, E., and Ghafoor, A. (2005). A trust-based context-aware access control model for web-services. *Distributed and Parallel Databases*, 18(1):83–105.

[51] Bhatti, R., Samuel, A., Eltabakh, M. Y., Amjad, H., and Ghafoor, A. (2007). Engineering a policy-based system for federated healthcare databases. *Knowledge and Data Engineering, IEEE Transactions on*, 19(9):1288–1304.

[52] Bhide, M. A. and Mohania, M. K. (2006). Event-based database access execution. *Google Patents*. US Patent 7,120,635.

[53] Bijon, K. Z., Krishnan, R., and Sandhu, R. (2013). A framework for risk-aware role based access control. In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pages 462–469. IEEE.

[54] Bishop, M., Engle, S., Frincke, D. A., Gates, C., Greitzer, F. L., Peisert, S., and Whalen, S. (2010). A risk management approach to the "insider threat". In *Insider threats in cyber security*, pages 115–137. Springer.

[55] Blackmer, W. (2016). Gdpr: Getting ready for the new eu general data protection regulation. *Information Law Group, InfoLawGroup LLP, Retrieved*, 22(08):2016.

[56] Blank, R. M. and Gallagher, P. D. (2012). Nist special publication 800-30: Guide for conducting risk assessments. *National Institute of Standards & Technology*.

[57] Blobel, B. (2004). Authorisation and access control for electronic health record systems. *International journal of medical informatics*, 73(3):251–257.

[58] Blumenthal, D. (2009). Stimulating the adoption of health information technology. *New England journal of medicine*, 360(15):1477–1479.

[59] Blumenthal, D. and Tavenner, M. (2010). The "meaningful use" regulation for electronic health records. *The New England Journal of Medicine*, 2010(363):501–504.

[60] Booch, G. (2005). *The unified modeling language user guide*. Pearson Education India.

[61] Boonstra, A., Versluis, A., and Vos, J. F. (2014). Implementing electronic health records in hospitals: a systematic literature review. *BMC health services research*, 14(1):370.

[62] Borrill, C., West, M., Shapiro, D., and Rees, A. (2000). Team working and effectiveness in health care. *British Journal of Healthcare Management*, 6(8):364–371.

[63] Bott, O. J. (2004). The electronic health record: Standardization and implementation. In *2nd OpenECG Workshop, Berlin, Germany*, pages 57–60.

[64] Braun, L. M., Wiesman, F., van den Herik, H. J., Hasman, A., and Korsten, E. (2007). Towards patient-related information needs. *International Journal of Medical Informatics*, 76(2):246–251.

[65] British Medical Association (2014). Access to health records: Guidance for health professionals in the united kingdom. Available from: `https://www.bma.org.uk/advice/employment/ethics/confidentiality-and-health-records/access-to-health-records`. Last accessed: August 2018.

[66] Buck, J., Garde, S., Kohl, C. D., and Knaup-Gregori, P. (2009). Towards a comprehensive electronic patient record to support an innovative individual care concept for premature infants using the openehr approach. *International journal of medical informatics*, 78(8):521–531.

[67] Budinsky, F. (2004). *Eclipse modeling framework: a developer's guide*. Addison-Wesley Professional.

[68] Buford, J. F., Lewis, L., and Jakobson, G. (2008). Insider threat detection using situation-aware mas. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE.

[69] Bureau, Federal Infrastructure Protection (2013). Unintentional insider threats: A foundational study. *Software Engineering Institute Technical Report*.

[70] Byrne, D. (2004). *Enabling Good Health for All : A Reflection Process for a New EU Health Strategy*. Commission of the European Communities. Available from:`http://ec.europa.eu/health/ph_overview/Documents/pub_good_health_en.pdf`. Last accessed: August 2018.

[71] Byun, J.-W., Bertino, E., and Li, N. (2005). Purpose based access control of complex data for privacy protection. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 102–110. ACM.

[72] Cabot, J. and Gogolla, M. (2012). Object constraint language (ocl): a definitive guide. In *Formal methods for model-driven engineering*, pages 58–90. Springer.

[73] Campbell, H., Hotchkiss, R., Bradshaw, N., and Porteous, M. (1998). Integrated care pathways. *BMJ: British Medical Journal*, 316(7125):133.

[74] Canada Health Infoway Inc (2005). Electronic health record infostructure (ehri) privacy and security conceptual architecture (version 1.1). Available from:`https://www.infoway-inforoute.ca/en/component/edocman/resources/technical-documents/387-ehr-privacy-and-security-architecture-full`. Last accessed: August 2018.

[75] Canada Health Infoway Inc (2008). A conceptual privacy impact assessment (pia) on canada's electronic health record solution (ehrs) blueprint version 2. Available from:`http://www.ehealthinformation.ca/`

wp-content/uploads/2014/08/pia.pdf. Last accessed: August 2018.

[76] Cánovas, Ó. and Gómez, A. F. (2003). Delegation in distributed systems: Challenges and open issues. In *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pages 499–503. IEEE.

[77] Cariou, E., Belloir, N., Barbier, F., and Djemam, N. (2010). Ocl contracts for the verification of model transformations. *Electronic Communications of the EASST*, 24.

[78] Catz, M. and Bayne, J. (2003). Canada health infoway–a pan-canadian approach. In *AMIA Annual Symposium Proceedings*, volume 2003, page 807. American Medical Informatics Association.

[79] Celikel, E., Kantarcioglu, M., Thuraisingham, B., and Bertino, E. (2009). A risk management approach to rbac. *Risk and Decision Analysis*, 1(1):21–33.

[80] Chao, C.-A. (2016). The impact of electronic health records on collaborative work routines: A narrative network analysis. *International journal of medical informatics*, 94:100–111.

[81] Chase, D. A., Ash, J. S., Cohen, D. J., Hall, J., Olson, G. M., and Dorr, D. A. (2014). The ehr's roles in collaboration between providers: A qualitative study. In *AMIA Annual Symposium Proceedings*, volume 2014, page 1718. American Medical Informatics Association.

[82] Chen, T.-Y., Chen, Y.-M., Chu, H.-C., and Chen, C.-C. (2007). Knowledge access control policy language model for virtual enterprises. In *Industrial Engineering and Engineering Management, 2007 IEEE International Conference on*, pages 1903–1907. IEEE.

[83] Chen, Y., Nyemba, S., and Malin, B. (2012). Detecting anomalous insiders in collaborative information systems. *Dependable and Secure Computing, IEEE Transactions on*, 9(3):332–344.

[84] Chen, Y., Nyemba, S., Zhang, W., and Malin, B. (2011). Leveraging social networks to detect anomalous insider actions in collaborative environments. In *Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on*, pages 119–124. IEEE.

[85] Cheng, P.-C., Rohatgi, P., Keser, C., Karger, P. A., Wagner, G. M., and Reninger, A. S. (2007). Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 222–230. IEEE.

[86] Chhanabhai, P. and Holt, A. (2007). Consumers are ready to accept the transition to online and electronic records if they can be assured of the security measures. *Medscape General Medicine*, 9(1):8.

[87] Christensen, B. and Ellingsen, G. (2016). Evaluating model-driven development for large-scale ehrs through the openehr approach. *International journal of medical informatics*, 89:43–54.

[88] Christiansen, E. K., Skipenes, E., Hausken, M. F., Skeie, S., østbye, T., and Iversen, M. M. (2017). Shared electronic health record systems: Key legal and security challenges. *Journal of Diabetes Science and Technology*, page 1932296817709797.

[89] Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., and Tacchella, A. (2002). Nusmv 2: An opensource tool for symbolic model checking. In *International Conference on Computer Aided Verification*, pages 359–364. Springer.

[90] Çınar, F. and Kaban, A. (2012). Conflict management and visionary leadership: An application in hospital organizations. *Procedia-Social and Behavioral Sciences*, 58:197–206.

[91] Clarke, M. A., Moore, J. L., Steege, L. M., Koopman, R. J., Belden, J. L., Canfield, S. M., Meadows, S. E., Elliott, S. G., and Kim, M. S. (2016). Health information needs, sources, and barriers of primary care patients to achieve patient-centered care: A literature review. *Health informatics journal*, 22(4):992–1016.

[92] Coorevits, P., Sundgren, M., Klein, G. O., Bahr, A., Claerhout, B., Daniel, C., Dugas, M., Dupont, D., Schmidt, A., Singleton, P., et al. (2013). Electronic health records: new opportunities for clinical research. *Journal of internal medicine*, 274(6):547–560.

[93] Córdoba, J.-R. and Piki, A. (2012). Facilitating project management education through groups as systems. *International Journal of Project Management*, 30(1):83–93.

[94] Covell, D. G., Uman, G. C., and Manning, P. R. (1985). Information needs in office practice: are they being met? *Annals of internal medicine*, 103(4):596–599.

[95] Cresswell, K. and Sheikh, A. (2009). The nhs care record service (nhs crs): recommendations from the literature on successful implementation and adoption. *Journal of Innovation in Health Informatics*, 17(3):153–160.

[96] Dahbur, K., Mohammad, B., and Tarakji, A. B. (2011). A survey of risks, threats and vulnerabilities in cloud computing. In *Proceedings of the 2011 International conference on intelligent semantic Web-services and applications*, page 12. ACM.

[97] Daiqin He, D. and Yang, J. (2009). Authorization control in collaborative healthcare systems. *Journal of theoretical and applied electronic commerce research*, 4(2):88–109.

[98] Dal Poz, M. R., Kinfu, Y., Dräger, S., Kunjumen, T., and Diallo, K. (2006). Counting health workers: definitions, data, methods and global results. *Geneva: World Health Organization*.

[99] Dekker, M. (2012). Critical cloud computing-a ciip perspective on cloud computing services. *Report of the European Network and Information Security Agency*.

[100] Del Fiol, G. and Haug, P. J. (2009). Classification models for the prediction of clinicians' information needs. *Journal of biomedical informatics*, 42(1):82–89.

[101] Del Fiol, G., Huser, V., Strasberg, H. R., Maviglia, S. M., Curtis, C., and Cimino, J. J. (2012). Implementations of the hl7 context-aware knowledge retrieval ("infobutton") standard: challenges, strengths, limitations, and uptake. *Journal of biomedical informatics*, 45(4):726–735.

[102] Deng, J.-B. and Hong, F. (2003). Task-based access control model. *Journal of Software*, 14(1):76–82.

[103] Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., and Young, P. R. (1989). Computing as a discipline. *Computer*, 22(2):63–70.

[104] Di Vimercati, S. D. C., Foresti, S., and Samarati, P. (2008). Recent advances in access control. In *Handbook of Database Security*, pages 1–26. Springer.

[105] Dimmock, N., Belokosztolszki, A., Eyers, D., Bacon, J., and Moody, K. (2004). Using trust and risk in role-based access control policies. In *Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 156–162. ACM.

[106] Doherty, R. B. and Crowley, R. A. (2013). Principles supporting dynamic clinical care teams: an american college of physicians position paper. *Annals of internal medicine*, 159(9):620–626.

[107] Dumortier, J. (2009). Study on the legal framework for interoperable e-health in europe. *Abschlussbericht: http://ec. europa. eu/information_society/activities/health/docs/studies/legal-fw-interop/E-Health-legal-fmwk-final-report. pdf*.

[108] Ekeland, A. G. (2016). Assessing electronic health records: Are basic assumptions in "health technology assessment" useful? In *Proceedings of the Eighth International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED 2016)*, pages 36–41.

[109] Elger, B. S., Iavindrasana, J., Iacono, L. L., Müller, H., Roduit, N., Summers, P., and Wright, J. (2010). Strategies for health data exchange for secondary, cross-institutional clinical research. *Computer methods and programs in biomedicine*, 99(3):230–251.

[110] Ely, J. W., Osheroff, J. A., Chambliss, M. L., Ebell, M. H., and Rosenbaum, M. E. (2005). Answering physicians' clinical questions: obstacles and potential solutions. *Journal of the American Medical Informatics Association*, 12(2):217–224.

[111] England NHS (2014). The nhs five year forward view. Available from: `https://www.england.nhs.uk/wp-content/uploads/2014/10/5yfv-web.pdf`. Last accessed: August 2018.

[112] Epstein, N. E. (2014). Multidisciplinary in-hospital teams improve patient outcomes: A review. *Surgical neurology international*, 5(Suppl 7):S295.

[113] Erickson, T. D., Kellogg, W. A., Malkin, P. K., Richards, J. T., and Philip, S. Y. (2003). Dynamic behavior-based access control system and method. US Patent 6,591,265.

[114] European Commission (1995). 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal of the European Communities*, 23(6).

[115] European Commission (2012). ehealth action plan 2012-2020 – innovative healthcare for the 21st century. *European Commission staff working document for informative purposes*. Available from:`https://ec.europa.eu/digital-single-market/en/news/ehealth-action-plan-2012-2020-innovative-healthcare`. Last accessed: August 2018.

[116] European Commission (2013). Overview of the national laws on electronic health records in the eu member states and their interaction with the provision of cross-border ehealth services. *EU Health Programme (2008-2013)*. Available from:`http://ec.europa.eu/health/ehealth/docs/laws_report_recommendations_en.pdf`. Last accessed: August 2018.

[117] European Commission (2015a). Commission report on the operation of directive 2011/24/eu on the application of patients' rights in cross-border healthcare. *Report from the commission to the european parliament and the council (Brussels- 2015)*. Available from: `http://ec.europa.eu/health/cross_border_care/policy/index_en.htm`. Last accessed: August 2018.

[118] European Commission (2015b). Expert panel on effective ways of investing in health: Cross-border cooperation. Available from: `http://ec.europa.eu/health/expert_panel/opinions/docs/009_crossborder_cooperation_en.pdf`. Last accessed: August 2018.

[119] Eysenbach, G. (2001). What is e-health? *Journal of medical Internet research*, 3(2):e20.

[120] Fabbri, D., LeFevre, K., and Hanauer, D. A. (2011). Explaining accesses to electronic health records. In *Proceedings of the 2011 Workshop on data mining for medicine and healthcare*, pages 10–17. ACM.

[121] Fabian, B., Ermakova, T., and Junghanns, P. (2015). Collaborative and secure sharing of healthcare data in multi-clouds. *Information Systems*, 48:132–150.

[122] Farzandipour, M., Sadoughi, F., Ahmadi, M., and Karimi, I. (2010). Security requirements and solutions in electronic health records: lessons learned from a comparative study. *Journal of medical systems*, 34(4):629–642.

[123] Feng, X., Ge, B., Sun, Y., Wang, Z., and Tang, D. (2010). Enhancing role management in role-based access control. In *Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on*, pages 677–683. IEEE.

[124] Fernández-Alemán, J. L., Señor, I. C., Lozoya, P. Á. O., and Toval, A. (2013). Security and privacy in electronic health records: A systematic literature review. *Journal of biomedical informatics*, 46(3):541–562.

[125] Ferraiolo, D., Chandramouli, R., Hu, V., and Kuhn, R. (2016). A comparison of attribute based access control (abac) standards for data service applications:extensible access control markup language (xacml) and next generation access control (ngac). *NIST Special Publication (800-178)*, 800(178). Available from: `http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-178.pdf`. Last accessed: August 2018.

[126] Ferraiolo, D., Cugini, J., and Kuhn, D. R. (1995). Role-based access control (rbac): Features and motivations. In *Proceedings of 11th annual computer security application conference*, pages 241–48.

[127] Ferraiolo, D., Kuhn, D. R., and Chandramouli, R. (2003). *Role-based access control*. Artech House.

[128] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R. (2001). Proposed nist standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):224–274.

[129] Ferreira, A., Ricardo, C.-C., Antunes, L., and Chadwick, D. (2007). Access control: how can it improve patients' healthcare? *Medical and Care Compunetics 4*, 4:65.

[130] Finn, R., Learmonth, M., and Reedy, P. (2010). Some unintended effects of teamwork in healthcare. *Social science & medicine*, 70(8):1148–1154.

[131] Firth-Cozens, J. (2001). Multidisciplinary teamwork: the good, bad, and everything in between. *BMJ Publishing Group Ltd*.

[132] Fleissig, A., Jenkins, V., Catt, S., and Fallowfield, L. (2006). Multidisciplinary teams in cancer care: are they effective in the uk? *The lancet oncology*, 7(11):935–943.

[133] Gajanayake, R., Iannella, R., and Sahama, T. (2011). Sharing with care: An information accountability perspective. *IEEE Internet Computing*, 15(4):31–38.

[134] Gajanayake, R., Iannella, R., and Sahama, T. (2014). Privacy oriented access control for electronic health records. *electronic Journal of Health Informatics*, 8(2):15.

[135] Gardner, D. B. (2005). Ten lessons in collaboration. *Online Journal of Issues in Nursing*, 10(1).

[136] Gartner, D., Kolisch, R., Neill, D. B., and Padman, R. (2015). Machine learning approaches for early drg classification and resource allocation. *INFORMS Journal on Computing*, 27(4):718–734.

[137] Gary, A., Debbie, F., and Melinda, A. (2004). A glossary of terms for community health care and services for older persons. Technical report, The World Health Organization (WHO).

[138] General Medical Council (2013). Delegation and referral. Available from:`https://www.gmc-uk.org/-/media/documents/delegation-and-referral_pdf-58834134.pdf`. Last accessed: Sep 2018.

[139] Georgiadis, C. K., Mavridis, I., Pangalos, G., and Thomas, R. K. (2001). Flexible team-based access control using contexts. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 21–27. ACM.

[140] Gerdes, M. and Fensli, R. (2015). End-to-end security and privacy protection for co-operative access to health and care data in a telehealth trial system for remote supervision of copd-patients. In *SHI 2015, Proceedings from The 13th Scandinavien Conference on Health Informatics, June 15-17, 2015, Tromsø, Norway*, number 115, pages 25–32. Linköping University Electronic Press.

[141] Gertz, M. and Jajodia, S. (2007). *Handbook of database security: applications and trends*. Springer Science & Business Media.

[142] Giokas, D. (2005). Canada health infoway-towards a national interoperable electronic health record (ehr) solution. *Studies in health technology and informatics*, 115:108–140.

[143] Giuri, L. and Iglio, P. (1997). Role templates for content-based access control. In *Proceedings of the second ACM workshop on Role-based access control*, pages 153–159. ACM.

[144] Gogolla, M. (2009). Object constraint language. In *Encyclopedia of Database Systems*, pages 1927–1929. Springer.

[145] Gorman, P. (2001). Information needs in primary care: a survey of rural and nonrural primary care physicians. *Medinfo*, 10(Pt 1):338–42.

[146] Gould, T., Crosby, D., Harmer, M., Lloyd, S., Lunn, J., Rees, G., Roberts, D., and Webster, J. (1992). Policy for controlling pain after surgery: effect of sequential changes in management. *Bmj*, 305(6863):1187–1193.

[147] Graetz, I., Huang, J., Brand, R., Shortell, S. M., Rundall, T. G., Bellows, J., Hsu, J., Jaffe, M., and Reed, M. E. (2015). The impact of electronic health records and teamwork on diabetes care quality. *The American journal of managed care*, 21(12):878.

[148] Graham, G. S. and Denning, P. J. (1972). Protection: principles and practice. In *Proceedings of the May 16-18, 1972, spring joint computer conference*, pages 417–429. ACM.

[149] Greitzer, F. L. and Hohimer, R. E. (2011). Modeling human behavior to anticipate insider attacks. *Journal of Strategic Security*, 4(2):25.

[150] Grimson, W., Berry, D., Grimson, J., Stephens, G., Felton, E., Given, P., and O'Moore, R. (1998). Federated healthcare record server—the synapses paradigm. *International Journal of Medical Informatics*, 52(1):3–27.

[151] Habib, M. A., Mahmood, N., Shahid, M., Aftab, M. U., Ahmad, U., and Faisal, C. M. N. (2014). Permission based implementation of dynamic separation of duty (dsd) in role based access control (rbac). In *Signal Processing and Communication Systems (ICSPCS), 2014 8th International Conference on*, pages 1–10. IEEE.

[152] Hafner, M., Memon, M., and Alam, M. (2007). Modeling and enforcing advanced access control policies in healthcare systems with sectet. In *Models in Software Engineering*, pages 132–144. Springer.

[153] Hajivali, M., Moghaddam, F. F., Alrashdan, M. T., and Alothmani, A. Z. (2013). Applying an agent-based user authentication and access control model for cloud servers. In *2013 International Conference on ICT Convergence (ICTC)*, pages 807–812.

[154] Hall, M. A. and Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *IEEE transactions on knowledge and data engineering*, 15(6):1437–1447.

[155] Han, K.-J., Kim, H. S., Kim, M.-J., Hong, K.-J., Park, S., Yun, S.-N., Song, M., Jung, Y., Kim, H., Kim, D.-O. D., et al. (2007). Thinking in clinical nursing practice: A study of critical care nurses' thinking applying the think-aloud, protocol analysis method. *Asian nursing research*, 1(1):68–82.

[156] Hansen, F. and Oleshchuk, V. (2003a). Application of role-based access control in wireless healthcare information systems. In *Scandinavian conference in health informatics*, pages 30–33.

[157] Hansen, F. and Oleshchuk, V. (2003b). Srbac: A spatial role-based access control model for mobile systems. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems (NORDSEC'03)*, pages 129–141.

[158] Harrison, M. A., Ruzzo, W. L., and Ullman, J. D. (1976). Protection in operating systems. *Communications of the ACM*, 19(8):461–471.

[159] Hartmann, S., Ma, H., et al. (2016). *27th International Conference Database and Expert Systems Applications, DEXA 2016, Porto, Portugal, September 5–8, 2016 Proceedings, Part I.* Springer.

[160] Hashem, Y., Takabi, H., GhasemiGol, M., and Dantu, R. (2016). Inside the mind of the insider: Towards insider threat detection using psychophysiological signals. *J. Internet Serv. Inf. Secur.*, 6(1):20–36.

[161] Häyrinen, K., Saranto, K., and Nykänen, P. (2008). Definition, structure, content, use and impacts of electronic health records: a review of the research literature. *International journal of medical informatics*, 77(5):291–304.

[162] He, Z., Wu, L., Li, H., Lai, H., and Hong, Z. (2011). Semantics-based access control approach for web service. *Journal of Computers*, 6(6):1152–1161.

[163] Healthcare Information and Management Systems Society (2014). Sixth annual himss security survey. Available from: `http://www.himss.org/2013-himss-security-survey`. Last accessed: August 2018.

[164] Heimly, V., Grimsmo, A., Faxvaag, A., et al. (2011). Diffusion of electronic health records and electronic communication in norway. *Applied clinical informatics*, 2(3):355–364.

[165] Heimly, V., Grimsmo, A., Henningsen, T. P., and Faxvaag, A. (2010). Diffusion and use of electronic health record systems in norway. *Studies in health technology and informatics*, 160(Pt 1):381.

[166] Helsedirektoratet (2017). The norwegian directorate of health. Available from: https://helsedirektoratet.no/, Last accessed: 2017-10-10.

[167] Hersh, W. (2004). Health care information technology: progress and barriers. *Jama*, 292(18):2273–2274.

[168] HL7 International. Health level seven international (hl7). Available from: `http://www.hl7.org/`. Last accessed: August 2018.

[169] Hofrichter, O., Gogolla, M., and Sohr, K. (2013). Uml/ocl based design and analysis of role-based access control policies. In *Joint Proceedings of the Workshops On the Globalization of Modeling Languages (GEMOC 2013) and Towards the Model Driven Organization*.

[170] Howard, M. and Lipner, S. (2006). *The security development lifecycle*, volume 8. Microsoft Press Redmond.

[171] Hu, V., Ferraiolo, D. F., Kuhn, D. R., Kacker, R. N., and Lei, Y. (2015). Implementing and managing policy rules in attribute based access control. In *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*, pages 518–525. IEEE.

[172] Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., et al. (2013). Guide to attribute based access control (abac) definition and considerations (draft). *NIST Special Publication*, 800:162.

[173] Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., and Scarfone, K. (2014). Guide to attribute based access control (abac) definition and considerations. *NIST Special Publication*, 800:162.

[174] Hu, V. C., Kuhn, D. R., Xie, T., and Hwang, J. (2011). Model checking for verification of mandatory access control models and properties. *International Journal of Software Engineering and Knowledge Engineering*, 21(01):103–127.

[175] Hu, V. C., Kuhn, R., and Yaga, D. (2017). Verification and test methods for access control policies/models. *NIST Special Publication*, 800:192.

[176] Hua, J. and Bapna, S. (2013). Who can we trust?: The economic impact of insider threats. *Journal of Global Information Technology Management*, 16(4):47–67.

[177] Humphreys, E. (2008). Information security management standards: Compliance, governance and risk management. *information security technical report*, 13(4):247–255.

[178] Hunker, J. and Probst, C. W. (2011). Insiders and insider threats-an overview of definitions and mitigation techniques. *JoWUA*, 2(1):4–27.

[179] Hwang, G.-H., Wu-Lee, C., and Jiang, Z.-X. (2012). Workflow-based dynamic access control in a service-oriented architecture. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, pages 47–52. IEEE.

[180] Hwang, J., Xie, T., Hu, V., and Altunay, M. (2010). Acpt: A tool for modeling and verifying access control policies. In *Policies for Distributed Systems and Networks (POLICY), 2010 IEEE International Symposium on*, pages 40–43. IEEE.

[181] Hystad, R. and Fensli, R. (2014). Access control for electronic health records. a delphi study of current challenges and highlighting of potential improvements. In *Scandinavian Conference on Health Informatics; August 22; 2014; Grimstad; Norway*, number 102, pages 37–44. Linköping University Electronic Press.

[182] Ikoev, V. (2016). Factors affecting the use of "kjernejournal" in the norwegian healthcare system. Master's thesis.

[183] Institute for critical infrastructure technology (2016). Hacking healthcare it in 2016 lessons the healthcare industry can learn from the opm breach. Available from: Last accessed: `http://icitech.org/hackinghealth16/`. August 2018.

[184] International Organization for Standardization (ISO) (2009). 73: 2009: Risk management vocabulary. *International Organization for Standardization, Geneva*. Available from: `https://www.iso.org/obp/ui/#iso:std:iso:guide:73:ed-1:v1:en`. Last accessed: August 2018.

[185] ITRC (2015). Identity theft resource centre (itrc) data breach reports. Available from: `http://www.idtheftcenter.org/images/breach/DataBreachReports_2015.pdf`. Last accessed: August 2018.

[186] Jaeger, T. and Prakash, A. (1996). Requirements of role-based access control for collaborative systems. In *Proceedings of the first ACM Workshop on Role-based access control*, page 16. ACM.

[187] Jafari, M., Safavi-Naini, R., Saunders, C., and Sheppard, N. P. (2010). Using digital rights management for securing data in a medical research environment. In *Proceedings of the tenth annual ACM workshop on Digital rights management*, pages 55–60. ACM.

[188] Jardim, S. V. (2013). The electronic health record and its contribution to healthcare information systems interoperability. *Procedia Technology*, 9:940–948.

[189] Java Platform (2017). The java tutorials: Package java.time. Available from: Last accessed: August 2018.

[190] Jha, A. K., Doolan, D., Grandt, D., Scott, T., and Bates, D. W. (2008). The use of health information technology in seven nations. *International journal of medical informatics*, 77(12):848–854.

[191] Jih, W.-R., Cheng, S.-y., Hsu, J. Y., Tsai, T.-M., et al. (2005). Context-aware access control in pervasive healthcare. *Computer Science and Information Engineering, National Taiwan University, Taiwan. jih@ agents. csie. ntu. edu. tw.,{r93070, yjhsu}@ csie. ntu. edu. tw*.

[192] Jin, X. (2014). *Attribute-based access control models and implementation in cloud infrastructure as a service*. The University of Texas at San Antonio.

[193] Jing, X., Liu, Z., Li, S., Qiao, B., and Tan, G. (2017). A cloud-user behavior assessment based dynamic access control model. *International Journal of System Assurance Engineering and Management*, 8(3):1966–1975.

[194] Jiong, Q. and Chen-hua, M. (2012). Detecting and resolving constraint conflicts in role-based access control. In *2011 International Conference on Electrical and Control Engineering*.

[195] Jnr, G. O. A. (2011). The effect of multidisciplinary team care on cancer management. *Pan African Medical Journal*, 9(1).

[196] Jonker, C., van Riemsdijk, M., and Vermeulen, B. (2011). Shared mental models. *Coordination, organizations, institutions, and norms in agent systems vi*, pages 132–151.

[197] Jonnalagadda, S. R., Del Fiol, G., Medlin, R., Weir, C., Fiszman, M., Mostafa, J., and Liu, H. (2012). Automatically extracting sentences from medline citations to support clinicians' information needs. *Journal of the American Medical Informatics Association*, 20(5):995–1000.

[198] Juhnke, C. (2012). Clinical and service integration. the route to improved outcomes.

[199] Kalra, D., Beale, T., and Heard, S. (2005). The openehr foundation. *Studies in health technology and informatics*, 115:153–173.

[200] Kandias, M., Virvilis, N., and Gritzalis, D. (2011). The insider threat in cloud computing. In *Critical Information Infrastructure Security*, pages 93–103. Springer.

[201] Kang, M. H., Park, J. S., and Froscher, J. N. (2001). Access control mechanisms for inter-organizational workflow. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 66–74. ACM.

[202] Kayem, A. V., Akl, S. G., and Martin, P. (2010). *Adaptive cryptographic access control*, volume 48. Springer Science & Business Media.

[203] Keller, M. E., Kelling, S. E., Cornelius, D. C., Oni, H. A., and Bright, D. R. (2015). Enhancing practice efficiency and patient care by sharing electronic health records. *Perspectives in health information management*, 12(Fall).

[204] Khambhammettu, H., Boulares, S., Adi, K., and Logrippo, L. (2012). A framework for threat assessment in access control systems. In *IFIP International Information Security Conference*, pages 187–198. Springer.

[205] Kikuchi, S., Tsuchiya, S., Adachi, M., and Katsuyama, T. (2007). Policy verification and validation framework based on model checking approach. In *Fourth International Conference on Autonomic Computing (ICAC'07)*, pages 1–1. IEEE.

[206] Kim, D.-K., Ray, I., France, R., and Li, N. (2004). Modeling role-based access control using parameterized uml models. In *International Conference on Fundamental Approaches to Software Engineering*, pages 180–193. Springer.

[207] Kim, M. M., Barnato, A. E., Angus, D. C., Fleisher, L. F., and Kahn, J. M. (2010). The effect of multidisciplinary care teams on intensive care unit mortality. *Archives of internal medicine*, 170(4):369–376.

[208] King, H. B., Battles, J., Baker, D. P., Alonso, A., Salas, E., Webster, J., Toomey, L., and Salisbury, M. (2008). Teamstepps™: team strategies and tools to enhance performance and patient safety.

[209] Kira, C. (2017). Insider-wrongdoing incident took 5+ years to discover. Available from: `https://post-healthcare.com` Last accessed: August 2018.

[210] Kizza, J. M. (2009). *Guide to computer network security*. Springer.

[211] Koch, M., Mancini, L. V., and Parisi-Presicce, F. (2002). Decidability of safety in graph-based models for access control. In *European Symposium on Research in Computer Security*, pages 229–244. Springer.

[212] Koien, G. M. and Oleshchuk, V. A. (2013). *Aspects of Personal Privacy in Communications: Problems, Technology and Solutions*. River Publishers.

[213] Koufi, V. and Vassilacopoulos, G. (2008). Context-aware access control for pervasive access to process-based healthcare systems. *Studies in health technology and informatics*, 136:679.

[214] Kruse, C. S., Kristof, C., Jones, B., Mitchell, E., and Martinez, A. (2016). Barriers to electronic health record adoption: a systematic literature review. *Journal of medical systems*, 40(12):252.

[215] Kugblenu, F. and Asim, M. (2007). Separation of duty in role based access control system: A case study. *Master's thesis, School of Engineering, Blekinge Institute of Technology*.

[216] Kuhlmann, M., Sohr, K., and Gogolla, M. (2013). Employing uml and ocl for designing and analysing role-based access control. *Mathematical Structures in Computer Science*, 23(4):796–833.

[217] Kuhn, D. R. (1997). Mutual exclusion of roles as a means of implementing separation of duty in role-based access control systems. In *Proceedings of the second ACM workshop on Role-based access control*, pages 23–30. ACM.

[218] Kuhn, D. R., Coyne, E. J., and Weil, T. R. (2010). Adding attributes to role-based access control. *IEEE Computer*, 43(6):79–81.

[219] Kuo, M.-H. (2011). Opportunities and challenges of cloud computing to improve health care services. *Journal of medical Internet research*, 13(3):e67.

[220] Lakkaraju, S. and Xu, D. (2014). Integrated modeling and analysis of attribute based access control policies and workflows in healthcare. In *Trustworthy Systems and their Applications (TSA), 2014 International Conference on*, pages 36–43. IEEE.

[221] Lawson, A. E. and Daniel, E. S. (2011). Inferences of clinical diagnostic reasoning and diagnostic error. *Journal of biomedical informatics*, 44(3):402–412.

[222] Le, X. H., Doll, T., Barbosu, M., Luque, A., and Wang, D. (2012). An enhancement of the role-based access control model to facilitate information access management in context of team collaboration and workflow. *Journal of biomedical informatics*, 45(6):1084–1107.

[223] Lei, Y., Kacker, R., Kuhn, D. R., Okun, V., and Lawrence, J. (2008). Ipog/ipog-d: efficient test generation for multi-way combinatorial testing. *Software Testing, Verification and Reliability*, 18(3):125–148.

[224] Li, J., Bai, Y., and Zaman, N. (2013). A fuzzy modeling approach for risk-based access control in ehealth cloud. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 17–23. IEEE.

[225] Li, N. and Tripunitara, M. V. (2006). Security analysis in role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 9(4):391–420.

[226] Li, N., Wang, Q., Qardaji, W., Bertino, E., Rao, P., Lobo, J., and Lin, D. (2009a). Access control policy combining: theory meets practice. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 135–144. ACM.

[227] Li, Q., Xu, M., and Zhang, X. (2008). Towards a group-based rbac model and decentralized user-role administration. In *28th International Conference on Distributed Computing Systems Workshops, 2008. ICDCS'08.*, pages 441–446. IEEE.

[228] Li, Q., Zhang, X., Xu, M., and Wu, J. (2009b). Towards secure dynamic collaborations with group-based rbac model. *computers & security*, 28(5):260–275.

[229] Lin, G., Wang, D., Bie, Y., and Lei, M. (2014). Mtbac: a mutual trust based access control model in cloud computing. *China Communications*, 11(4):154–162.

[230] Liu, A. X., Chen, F., Hwang, J., and Xie, T. (2011a). Designing fast and scalable xacml policy evaluation engines. *Computers, IEEE Transactions on*, 60(12):1802–1817.

[231] Liu, L. S., Shih, P. C., and Hayes, G. R. (2011b). Barriers to the adoption and use of personal health record systems. In *Proceedings of the 2011 iConference*, pages 363–370. ACM.

[232] Liu, V., Musen, M. A., and Chou, T. (2015). Data breaches of protected health information in the united states. *The Journal of the American Medical Association (JAMA)*, 313(14):1471–1473.

[233] Ma, J. (2012). A formal approach for risk assessment in rbac systems. *Journal of Universal Computer Science*, 18(17):2432–2451.

[234] Ma, S. and Wang, Y. (2013). Self-adaptive access control model based on feedback loop. In *Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on*, pages 597–602. IEEE.

[235] Ma, T., Wang, H., Cao, J., Yong, J., and Zhao, Y. (2016). Access control management with provenance in healthcare environments. In *Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on*, pages 545–550. IEEE.

[236] Mahmudlu, R., den Hartog, J., and Zannone, N. (2016). Data governance and transparency for collaborative systems. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 199–216. Springer.

[237] Majumder, A., Namasudra, S., and Nath, S. (2014). Taxonomy and classification of access control models for cloud environments. In *Continued Rise of the Cloud*, pages 23–53. Springer.

[238] Martin, E., Hwang, J., Xie, T., and Hu, V. (2008). Assessing quality of policy properties in verification of access control policies. In *Computer Security Applications Conference, 2008. ACSAC 2008. Annual*, pages 163–172. IEEE.

[239] Martin, E., Xie, T., and Yu, T. (2006). Defining and measuring policy coverage in testing access control policies. In *International Conference on Information and Communications Security*, pages 139–158. Springer.

[240] Mattingly, C. (1991). What is clinical reasoning? *American Journal of Occupational Therapy*, 45(11):979–986.

[241] Maviglia, S. M., Yoon, C. S., Bates, D. W., and Kuperman, G. (2006). Knowledgelink: impact of context-sensitive information retrieval on clinicians' information needs. *Journal of the American Medical Informatics Association*, 13(1):67–73.

[242] Maw, H. A., Xiao, H., and Christianson, B. (2013). An adaptive access control model for medical data in wireless sensor networks. In *e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on*, pages 303–309. IEEE.

[243] Mayer, R. C., Davis, J. H., and Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of management review*, 20(3):709–734.

[244] Meingast, M., Roosta, T., and Sastry, S. (2006). Security and privacy issues with health care information technology. In *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2006. EMBS'06.*, pages 5453–5458. IEEE.

[245] Menachemi, N. and Collum, T. H. (2011). Benefits and drawbacks of electronic health record systems. *Risk Management and Healthcare Policy*, 4:47–55.

[246] Mental Health Commission and others (2006). Multidisciplinary team working: from theory to practice. *Dublin: Mental Health Commission*.

[247] Meredith Belbin, R. (2010). Management teams: Why they succeed or fail. *Human Resource Management International Digest*, 19(3).

[248] Meslec, N. and Curşeu, P. L. (2015). Are balanced groups better? belbin roles in collaborative learning groups. *Learning and Individual Differences*, 39:81–88.

[249] Mickan, S. M. (2005). Evaluating the effectiveness of health care teams. *Australian Health Review*, 29(2):211–217.

[250] Mickan, S. M. and Rodger, S. A. (2005). Effective health care teams: a model of six characteristics developed from shared perceptions. *Journal of interprofessional care*, 19(4):358–370.

[251] Moffett, J. D. (1994). Specification of management policies and discretionary access control. *Network and distributed systems management*, pages 455–480.

[252] Mohammad, A., Kanaan, G., Khdour, T., and Bani-Ahmad, S. (2011a). Ontology-based access control model for semantic web service. *Journal of Information and Computing Science*, 6(3):177–194.

[253] Mohammad, A., Khdour, T., Kanaan, G., Kanaan, R., and Ahmad, S. (2011b). Analysis of existing access control models from web services applications' perspective. *J. Comput*, 3:10–16.

[254] Monteleone, P. P., Rosenfield, K., and Rosovsky, R. P. (2016). Multidisciplinary pulmonary embolism response teams and systems. *Cardiovascular diagnosis and therapy*, 6(6):662.

[255] Moonian, O., Cheerkoot-Jalim, S., Nagowah, S. D., Khedo, K. K., Doomun, R., and Cadersaib, Z. (2008). Hcrbac–an access control system for collaborative context-aware healthcare services in mauritius. *Journal of Health Informatics in Developing Countries*, 2(2).

[256] Motta, G. H. and Furuie, S. S. (2003). A contextual role-based access control authorization model for electronic patient record. *Information Technology in Biomedicine, IEEE Transactions on*, 7(3):202–207.

[257] Myers, A. C. and Liskov, B. (1997). *A decentralized model for information flow control*, volume 31. ACM.

[258] Namasudra, S., Nath, S., and Majumder, A. (2014). Profile based access control model in cloud computing environment. In *Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on*, pages 1–5. IEEE.

[259] Narayan, S., Gagné, M., and Safavi-Naini, R. (2010). Privacy preserving ehr system using attribute-based infrastructure. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pages 47–52. ACM.

[260] National Patient Safety Agency (2008). Risk matrix for risk managers. Available from:`http://www.nrls.npsa.nhs.uk/EasySiteWeb/getresource.axd?AssetID=60149`. Last accessed: August 2018.

[261] Ngo, C., Demchenko, Y., and de Laat, C. (2015). Decision diagrams for xacml policy evaluation and management. *Computers & Security*, 49:1–16.

[262] Nguyen, N. T., Reiher, P. L., and Kuenning, G. H. (2003). Detecting insider threats by monitoring system call activity. In *IAW*, pages 45–52. Citeseer.

[263] Ni, Q., Bertino, E., and Lobo, J. (2010). Risk-based access control systems built on fuzzy inferences. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 250–260. ACM.

[264] Northouse, P. G. (2013). *Leadership: Theory and practice*. Sage publications.

[265] Norwegian Directorate of eHealth (2017). Code of conduct for information security the healthcare and care services sector. *The Norwegian Directorate of eHealth (NDE)*. Available from: `https://ehelse.no/english`. Last accessed: August 2018.

[266] Norwegian Ministry of Health and Care Services. The coordination reform, proper treatment - at the right place and right time. Available from: Last accessed: 2017-09-30.

[267] Norwegian Ministry of Health and Care Services. National health and care services plan (2011–2015). Available from: `https://www.regjeringen.no/contentassets/f17befe0cb4c48d68c744bce3673413d/en-gb/pdfs/stm201020110016000en_pdfs.pdf`. Last accessed: August 2018.

[268] Nosowsky, R. and Giordano, T. J. (2006). The health insurance portability and accountability act of 1996 (hipaa) privacy rule: implications for clinical research. *Annu. Rev. Med.*, 57:575–590.

[269] OASIS XACML Technical Committee (2013). extensible access control markup language (xacml) version 3.0. *Oasis standard, OASIS*. Available from: `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html`. Last accessed: August 2018.

[270] Oh, S. and Park, S. (2003). Task–role-based access control model. *Information systems*, 28(6):533–562.

[271] Omg, O. and Specification, Q. F. A. (2007). Object management group. *Home page: http://www. omg. org*.

[272] Omicini, A., Ricci, A., and Viroli, M. (2005). Rbac for organisation and security in an agent coordination infrastructure. *Electronic Notes in Theoretical Computer Science*, 128(5):65–85.

[273] openEHR. An open domain-driven platform for developing flexible e-health systems. Available from:`http://www.openehr.org/`. Last accessed: August 2018.

[274] Ozair, F. F., Jamshed, N., Sharma, A., and Aggarwal, P. (2015). Ethical issues in electronic health records: A general overview. *Perspectives in clinical research*, 6(2):73.

[275] Paci, F., Squicciarini, A., and Zannone, N. (2018). Survey on access control for community-centered collaborative systems. *ACM Computing Surveys (CSUR)*, 51(1):6.

[276] Park, J. and Sandhu, R. (2004). The ucon abc usage control model. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):128–174.

[277] Parkin, E. (2016). Patient health records and confidentiality. *House of commons library (briefing paper)*, (07103). Available from:`https://beta.parliament.uk/search?q=Patient+health+records+and+confidentiality`. Last accessed: August 2018.

[278] Passarani, I. (2013). Patient access to electronic health records. *Report of the eHealth Stakeholder Group*. Available from:`http://ec.europa.eu/health/expert_panel/opinions/docs/009_crossborder_cooperation_en.pdf`. Last accessed: August 2018.

[279] Peng, H., Gu, J., and Ye, X. (2008). Dynamic purpose-based access control. In *Parallel and Distributed Processing with Applications, 2008. ISPA'08. International Symposium on*, pages 695–700. IEEE.

[280] Perkinson, R. R. (2016). *Chemical dependency counseling: A practical guide*. Sage Publications. Available from:`https://www.sagepub.com/sites/default/files/upm-binaries/18970_Chapter_5.pdf`. Last accessed: August 2018.

[281] Petri, L. (2010). Concept analysis of interdisciplinary collaboration. In *Nursing forum*, volume 45, pages 73–82. Wiley Online Library.

[282] Poissant, L., Pereira, J., Tamblyn, R., and Kawasumi, Y. (2005). The impact of electronic health records on time efficiency of physicians and nurses: a systematic review. *Journal of the American Medical Informatics Association*, 12(5):505–516.

[283] Prados-Suárez, B., Molina, C., Yañez, C. P., and de Reyes, M. P. (2012). Improving electronic health records retrieval using contexts. *Expert Systems with Applications*, 39(10):8522–8536.

[284] Probst, C. W., Hunker, J., Gollmann, D., and Bishop, M. (2010). *Insider Threats in Cyber Security*, volume 49. Springer Science & Business Media.

[285] Public Law (2009). 111–5–american recovery and reinvestment act of 2009. *US Government Printing Office, Washington, DC (February 2009)*. Available from: `https://www.gpo.gov/fdsys/pkg/PLAW-111publ5/pdf/PLAW-111publ5.pdf`. Last accessed: August 2018.

[286] Pulur, N. A., Altop, D. K., and Levi, A. (2016). A role and activity based access control for secure healthcare systems. In *Information Sciences and Systems 2015*, pages 93–103. Springer.

[287] Qiao, L., Li, Y., Chen, X., Yang, S., Gao, P., Liu, H., Feng, Z., Nian, Y., and Qiu, M. (2015). Medical high-resolution image sharing and electronic whiteboard system: A pure-web-based system for accessing and discussing lossless original images in telemedicine. *Computer methods and programs in biomedicine*, 121(2):77–91.

[288] Quinn, K. S. (2010). Computer crime and security survey. *New Zealand*.

[289] Qureshi, S. P., Rankin, K., Storrar, N., and Freeman, M. Preparation for making clinical referrals. *The clinical teacher*.

[290] Rajkomar, A. and Dhaliwal, G. (2011). Improving diagnostic reasoning to improve patient safety. *The Permanente Journal*, 15(3):68.

[291] Raman, R. S., Jagannathan, V., and Reddy, R. (1997). Secure collaboration technology for healthcare enterprises. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1997. Proceedings., Sixth IEEE Workshops on*, pages 263–268. IEEE.

[292] Ramli, C. D. P. K. (2015). *Modelling and Analysing Access Control Policies in XACML 3.0*. PhD thesis, Technical University of Denmark.

[293] Ravari, A. N., Amini, M., Jalili, R., and Jafarian, J. H. (2008). A history based semantic aware access control model using logical time. In *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*, pages 43–50. IEEE.

[294] Ray, I., Li, N., France, R., and Kim, D.-K. (2004). Using uml to visualize role-based access control constraints. In *Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 115–124. ACM.

[295] Redfern, E., Brown, R., and Vincent, C. (2009). Improving communication in the emergency department. *Emergency Medicine Journal*, 26(9):658–661.

[296] Redspin (2016). Breach report 2015: Protected health information (phi). Available from:`https://www.redspin.com/resources/download/breach-report-2015-protected-health-information-phi/`. Last accessed: 2017-09-30.

[297] Rees, C. E. and Bath, P. A. (2000). The information needs and source preferences of women with breast cancer and their family members: a review of the literature published between 1988 and 1998. *Journal of advanced nursing*, 31(4):833–841.

[298] Reis, F. F., Costa-Pereira, A., and Correia, M. E. (2008). Access and privacy rights using web security standards to increase patient empowerment. *Studies in health technology and informatics*, 137:275–285.

[299] Reitz, R., Common, K., Fifield, P., and Stiasny, E. (2012). Collaboration in the presence of an electronic health record. *Families, Systems, & Health*, 30(1):72.

[300] Rittenberg, L. and Martens, F. (2012). Enterprise risk management: understanding and communicating risk appetite. *Committee of Sponsoring Organizations of the Treadway Commission (COSO)*.

[301] Rizvi, S., Mendelzon, A., Sudarshan, S., and Roy, P. (2004). Extending query rewriting techniques for fine-grained access control. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 551–562. ACM.

[302] Ross, R. S. (2011). Managing information security risk: Organization, mission, and information system view. *Special Publication (NIST SP)-800-39*.

[303] Røstad, L. (2008a). *Access control in healthcare information systems*. PhD thesis, Norwegian University of Science and Technology.

[304] Røstad, L. (2008b). An initial model and a discussion of access control in patient controlled health records. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 935–942. IEEE.

[305] Rostad, L. and Edsberg, O. (2006). A study of access control requirements for healthcare systems based on audit trails from access logs. In *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*, pages 175–186. IEEE.

[306] Rostad, L., Nytro, O., Tondel, I., and Meland, P. H. (2007). Access control and integration of health care systems: An experience report and future challenges. In *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pages 871–878. IEEE.

[307] Rothstein, M. A. (2013). Hipaa privacy rule 2.0. *The Journal of Law, Medicine & Ethics*, 41(2):525–528.

[308] Ruan, C. and Varadharajan, V. (2002). Resolving conflicts in authorization delegations. In *Australasian Conference on Information Security and Privacy*, pages 271–285. Springer.

[309] Rubio-Medrano, C. E., D'Souza, C., and Ahn, G.-J. (2013). Supporting secure collaborations with attribute-based access control. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, pages 525–530. IEEE.

[310] Russello, G., Dong, C., and Dulay, N. (2008). A workflow-based access control framework for e-health applications. In *AINAW 2008-Workshops. 22nd International Conference on*, pages 111–120. IEEE.

[311] Salem, M. B., Hershkop, S., and Stolfo, S. J. (2008). A survey of insider attack detection research. *Insider Attack and Cyber Security*, pages 69–90.

[312] Salim, F., Reid, J., and Dawson, E. (2010). Authorization models for secure information sharing: A survey and research agenda. *The ISC International Journal of Information Security*, 2(2):69–87.

[313] Salim, F., Reid, J., Dawson, E., and Dulleck, U. (2011). An approach to access control under uncertainty. In *Availability, reliability and security (ARES), 2011 Sixth International conference on*, pages 1–8. IEEE.

[314] Samarati, P. and de Vimercati, S. C. (2001). Access control: Policies, models, and mechanisms. In *International School on Foundations of Security Analysis and Design*, volume 2171, pages 137–196. Springer.

[315] Sandhu, R., Ferraiolo, D., and Kuhn, R. (2000). The nist model for role-based access control: towards a unified standard. In *ACM workshop on Role-based access control*, volume 2000, pages 1–17.

[316] Sandhu, R. S. (1993). Lattice-based access control models. *Computer*, 26(11):9–19.

[317] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. (1996). Role-based access control models. *Computer*, (2):38–47.

[318] Schrijvers, G., van Hoorn, A., and Huiskes, N. (2012). The care pathway: concepts and theories: an introduction. *International Journal of Integrated Care*, 12(Special Edition Integrated Care Pathways).

[319] Schultz, D. (2012). Medical data breaches raising alarm. *Washington Post*. Available from: `https://www.washingtonpost.com/national/health-science/medical-data-breaches-raise-alarms/2012/06/02/gJQAVPWt9U_story.html?utm_term=.bd85bf8296ff`. Last accessed: August 2018.

[320] Schwartmann, D. (2004). An attributable role-based access control for healthcare. In *Computational Science-ICCS 2004*, pages 1148–1155. Springer.

[321] Shah, C. (2010). A framework for supporting user-centric collaborative information seeking. In *PhD Thesis. University of North Carolina*, pages 1–268. Available from: `http://comminfo.rutgers.edu/~chirags/papers/Shah_Dissertation.pdf`. Last accessed: August 2018.

[322] Shaikh, R. A., Adi, K., and Logrippo, L. (2012). Dynamic risk-based decision methods for access control systems. *computers & security*, 31(4):447–464.

[323] Shaikh, R. A., Adi, K., and Logrippo, L. (2016). A data classification method for inconsistency and incompleteness detection in access control policy sets. *International Journal of Information Security*, pages 1–23.

[324] Shaikh, R. A., Adi, K., Logrippo, L., and Mankovski, S. (2010a). Detecting incompleteness in access control policies using data classification schemes. In *Digital Information Management (ICDIM), 2010 Fifth International Conference on*, pages 417–422. IEEE.

[325] Shaikh, R. A., Adi, K., Logrippo, L., and Mankovski, S. (2010b). Inconsistency detection method for access control policies. In *Information Assurance and Security (IAS), 2010 Sixth International Conference on*, pages 204–209. IEEE.

[326] Shaikh, R. A., Adi, K., Logrippo, L., and Mankovski, S. (2011). Risk-based decision method for access control systems. In *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*, pages 189–192. IEEE.

[327] Sharma, M., Bai, Y., Chung, S., and Dai, L. (2012). Using risk in access control for cloud-assisted ehealth. In *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on*, pages 1047–1052. IEEE.

[328] Sharmin, M., Ahmed, S., and Ahamed, S. I. (2005). Safe-rd (secure, adaptive, fault tolerant, and efficient resource discovery) in pervasive computing environments. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 2, pages 271–276. IEEE.

[329] Shea-Budgell, M., Kostaras, X., Myhill, K., and Hagen, N. (2014). Information needs and sources of information for patients during cancer follow-up. *Current oncology*, 21(4):165.

[330] Sheikh, A., Cornford, T., Barber, N., Avery, A., Takian, A., Lichtner, V., Petrakaki, D., Crowe, S., Marsden, K., Robertson, A., et al. (2011). Implementation and adoption of nationwide electronic health records in secondary care in england: final qualitative results from prospective national evaluation in "early adopter" hospitals. *Bmj*, 343:d6054.

[331] Shen, H. and Dewan, P. (1992). Access control for collaborative environments. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 51–58. ACM.

[332] Shin, M. E. and Ahn, G.-J. (2000). Uml-based representation of role-based access control. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proeedings. IEEE 9th International Workshops on*, pages 195–200. IEEE.

[333] Shoniregun, C. A., Dube, K., and Mtenzi, F. (2010). *Electronic healthcare information security*, volume 53. Springer Science & Business Media.

[334] Shouval, R., Bondi, O., Mishan, H., Shimoni, A., Unger, R., and Nagler, A. (2014). Application of machine learning algorithms for clinical predictive modeling: a data-mining approach in sct. *Bone marrow transplantation*, 49(3):332.

[335] Silow-Carroll, S., Edwards, J. N., and Rodin, D. (2012). Using electronic health records to improve quality and efficiency: the experiences of leading hospitals. *Issue Brief (The Commonwealth Fund)*, 17:1–40.

[336] Sinclair, S. and Smith, S. W. (2008). Preventative directions for insider threat mitigation via access control. In *Insider Attack and Cyber Security*, pages 165–194. Springer.

[337] Singh, D. and Ham, C. (2006). *Improving care for people with long-term conditions: a review of UK and international frameworks*. University of Birmingham. Health services management centre.

[338] Smaradottir, B., Gerdes, M., Martinez, S., and Fensli, R. (2016a). The eu-project united4health: User-centred design of an information system for a norwegian telemedicine service. *Journal of telemedicine and telecare*, 22(7):422–429.

[339] Smaradottir, B., Martinez, S., Holen-Rabbersvik, E., Vatnøy, T. K., and Fensli, R. W. (2016b). Usability evaluation of a collaborative health information system. lessons from a user-centred design process. In *HEALTHINF'16, 9th International Conference on Health Informatics*, volume 5, pages 306–313.

[340] Smaradottir, B. F., Martinez, S., Holen-Rabbersvik, E., and Fensli, R. (2015). ehealth-extended care coordination: Development of a collaborative system for inter-municipal dementia teams: A research project with a user-centered design approach. In *International Conference on Computational Science and Computational Intelligence (CSCI2015)*, pages 749–753. IEEE.

[341] Smith, R. (1996). What clinical information do doctors need? *Bmj*, 313(7064):1062–1068.

[342] Snell, E. (2017). Healthcare data breach costs highest for 7th straight year. Available from: Last accessed: August 2018.

[343] Sohr, K., Ahn, G.-J., Gogolla, M., and Migge, L. (2005). Specification and validation of authorisation constraints using uml and ocl. In *Computer Security–ESORICS 2005*, pages 64–79. Springer.

[344] Sohr, K., Drouineaud, M., Ahn, G.-J., and Gogolla, M. (2008a). Analyzing and managing role-based access control policies. *IEEE Transactions on Knowledge and Data Engineering*, 20(7):924–939.

[345] Sohr, K., Mustafa, T., Bao, X., and Ahn, G.-J. (2008b). Enforcing role-based access control policies in web services with uml and ocl. In *Computer Security Applications Conference, 2008. ACSAC 2008. Annual*, pages 257–266. IEEE.

[346] Steinberg, D., Budinsky, F., Merks, E., and Paternostro, M. (2008). *EMF: eclipse modeling framework*. Pearson Education.

[347] Steinbrook, R. (2009). Health care and the american recovery and reinvestment act. *New England Journal of Medicine*, 360(11):1057–1060.

[348] Strembeck, M. and Mendling, J. (2011). Modeling process-related rbac models with extended uml activity models. *Information and Software Technology*, 53(5):456–483.

[349] Stroetmann, K. A., Artmann, J., Dumortier, J., and Verhenneman, G. (2012). United in diversity: legal challenges on the road towards interoperable ehealth solutions in europe. *EJBI*, 8(2):3–10.

[350] Stroetmann, K. A., Artmann, J., Stroetmann, V. N., Protti, D., Dumortier, J., Giest, S., Walossek, U., and Whitehouse, D. (2011). European countries on their journey towards national ehealth infrastructures. *Luxembourg: Office for Official Publications of the European Communities*.

[351] Sujansky, W. (2001). Heterogeneous database integration in biomedicine. *Journal of biomedical informatics*, 34(4):285–298.

[352] Sun, L., Wang, H., Soar, J., and Rong, C. (2012). Purpose based access control for privacy protection in e-healthcare services. *Journal of Software*, 7(11):2443–2449.

[353] Taylor, C., Munro, A. J., Glynne-Jones, R., Griffith, C., Trevatt, P., Richards, M., and Ramirez, A. J. (2010). Multidisciplinary team working in cancer: what is the evidence? *BMJ*, 340:c951.

[354] Thakore, D. (2013). Conflict and conflict management. *IOSR Journal of Business and Management (IOSR-JBM)*, 8(6).

[355] Thomas, J. et al. (2009). Medical records and issues in negligence. *Indian Journal of Urology*, 25(3):384.

[356] Thomas, R. K. (1997). Team-based access control (tmac): a primitive for applying role-based access controls in collaborative environments. In *Proceedings of the second ACM workshop on Role-based access control*, pages 13–19. ACM.

[357] Thomas, R. K. and Sandhu, R. S. (1998). Task-based authorization controls (tbac): A family of models for active and enterprise-oriented authorization management. In *Database Security XI*, pages 166–181. Springer.

[358] Tikkinen-Piri, C., Rohunen, A., and Markkula, J. (2018). Eu general data protection regulation: Changes and implications for personal data collecting companies. *Computer Law & Security Review*, 34(1):134–153.

[359] Tolone, W., Ahn, G.-J., Pai, T., and Hong, S.-P. (2005). Access control in collaborative systems. *ACM Computing Surveys (CSUR)*, 37(1):29–41.

[360] Ubale Swapnaja, A., Modani Dattatray, G., and Apte Sulabha, S. (2014). Analysis of dac mac rbac access control based models for security. *International Journal of Computer Applications*, 104(5).

[361] UK Gov (2016). Electronic health records. Availbale: `http://researchbriefings.parliament.uk/ResearchBriefing/Summary/POST-PN-0519`. Last accessed: August 2018.

[362] Ulltveit-Moe, N. and Oleshchuk, V. (2016). Enforcing mobile security with location-aware role-based access control. *Security and Communication Networks*, 9(5):429–439.

[363] United Kingdom General Medical Council (2013). Good medical practice. Available from:`http://www.gmc-uk.org/guidance/`. Last accessed: August 2018.

[364] United Kingdom General Medical Council (2017). Confidentiality: good practice in handling patient information. Available from: `https://www.gmc-uk.org/-/media/documents/Confidentiality_good_practice_in_handling_patient_information___English_0417.pdf_70080105.pdf`. Last accessed: 12-01-2018.

[365] United4Health (2017). European commission competitiveness innovation programme. Available from: `http://www.united4health.eu`. Last accessed August 2018S.

[366] US Centers for Disease Control and Prevention (2003). Hipaa privacy rule and public health. guidance from cdc and the us department of health and human services. *MMWR: Morbidity and mortality weekly report*, 52(Suppl. 1):1–17.

[367] US Department of Health and Human Services. Hipaa privacy rule and sharing information related to mental health. Available from:`http://www.hhs.gov/hipaa/for-professionals/special-topics/mental-health/`. Last accessed: August 2018.

[368] US Department of Health and Human Services et al. (2017). Health care industry cybersecurity task force. Available from: `https://www.phe.gov/Preparedness/planning/CyberTF/Documents/report2017.pdf`. Last accessed: August 2018.

[369] van der Linden, H., Kalra, D., Hasman, A., and Talmon, J. (2009). Inter-organizational future proof ehr systems: a review of the security and privacy related issues. *International journal of medical informatics*, 78(3):141–160.

[370] Vanhaecht, K., Panella, M., Van Zelm, R., and Sermeus, W. (2011). What about care pathways. *Care of Dying. A pathway to excellence*, pages 2–12.

[371] Vawdrey, D. K., Wilcox, L. G., Collins, S., Feiner, S., Mamykina, O., Stein, D. M., Bakken, S., Fred, M. R., Stetson, P. D., et al. (2011). Awareness of the care team in electronic health records. *Appl Clin Inform*, 2(4):395–405.

[372] Verma, S., Kumar, S., and Singh, M. (2012). Comparative analysis of role base and attribute base access control model in semantic web. *International Journal of Computer Applications*, 46(18).

[373] Vicente, C. R., Kirkpatrick, M., Ghinita, G., Bertino, E., and Jensen, C. S. (2009). Towards location-based access control in healthcare emergency response. In *Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*, pages 22–26. ACM.

[374] Vincent, C. H., Ferraiolo, D., and Kuhn, D. R. (2006). Assessment of access control systems. *Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD*, pages 20899–8930.

[375] Vodicka, E., Mejilla, R., Leveille, S. G., Ralston, J. D., Darer, J. D., Delbanco, T., Walker, J., and Elmore, J. G. (2013). Online access to doctors' notes: patient concerns about privacy. *Journal of medical Internet research*, 15(9).

[376] Voigt, P. and von dem Bussche, A. (2017). *The Eu General Data Protection Regulation (gdpr): A Practical Guide*. Springer.

[377] Wang, H., Sun, L., and Varadharajan, V. (2010). Purpose-based access control policies and conflicting analysis. In *Security and Privacy–Silver Linings in the Cloud*, pages 217–228. Springer.

[378] Wang, Q. and Jin, H. (2011). Quantified risk-adaptive access control for patient privacy protection in health information systems. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 406–410. ACM.

[379] Wang, W. (1999). Team-and-role-based organizational context and access control for cooperative hypermedia environments. In *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: returning to our diverse roots*, pages 37–46. ACM.

[380] Wang, X. and Rutle, A. (2014). Model checking healthcare workflows using alloy. *Procedia Computer Science*, 37:481–488.

[381] Warmer, J. B. and Kleppe, A. G. (1998). The object constraint language: Precise modeling with uml (addison-wesley object technology series).

[382] Warmer, J. B. and Kleppe, A. G. (2003). *The object constraint language: getting your models ready for MDA*. Addison-Wesley Professional.

[383] Watkins, B. (2014). The impact of cyber attacks on the private sector. *Briefing Paper, Association for International Affair*, page 12.

[384] Weller, J., Boyd, M., and Cumin, D. (2014). Teams, tribes and patient safety: overcoming barriers to effective teamwork in healthcare. *Postgraduate medical journal*, 90(1061):149–154.

[385] Wen, Z., Zhou, B., and Wu, D. (2009). Three-layers role-based access control framework in large financial web systems. In *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, pages 1–4. IEEE.

[386] Westli, H. K., Johnsen, B. H., Eid, J., Rasten, I., and Brattebø, G. (2010). Teamwork skills, shared mental models, and performance in simulated trauma teams: an independent group design. *Scandinavian journal of trauma, resuscitation and emergency medicine*, 18(1):47.

[387] Wikborg, R. and Co, A. D. (2014). Overview of the national laws on electronic health records in the eu member states. national report for norway. Available from: `https://ec.europa.eu/health/ehealth/projects/`

nationallaws_electronichealthrecords_en. Last accessed: August 2018.

[388] William, S. and Brown, L. (2014). *Computer Security: Principles And Practice, Global Edition*. Pearson Education Limited.

[389] Winter, A., Haux, R., Ammenwerth, E., Brigl, B., Hellrung, N., and Jahn, F. (2010). Health information systems. In *Health Information Systems*, pages 33–42. Springer.

[390] Wismar, M., Palm, W., Figueras, J., Ernst, K., Van Ginneken, E., et al. (2011). Cross-border health care in the european union: mapping and analysing practices and policies. *World Health Organization*.

[391] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[392] Wong, H. J., Caesar, M., Bandali, S., Agnew, J., and Abrams, H. (2009). Electronic inpatient whiteboards: improving multidisciplinary communication and coordination of care. *International Journal of Medical Informatics*, 78(4):239–247.

[393] World Health Organization (2010). Classifying health workers: Mapping occupations to the international standard classification. *Geneva: WHO*. Available from: http://www.who.int/hrh/statistics/Health_workers_classification.pdf. Last accessed: August 2018.

[394] World Health Organization (2016). Integrated care models: an overview. *Working document. Copenhagen (DK): WHO Regional Office for Europe*. Available from: http://www.euro.who.int/__data/assets/pdf_file/0005/322475/Integrated-care-models-overview.pdf. Last accessed: August 2018.

[395] Wu, R., Ahn, G.-J., and Hu, H. (2012). Secure sharing of electronic health records in clouds. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, pages 711–718. IEEE.

[396] Xiao, L., Hu, B., Croitoru, M., Lewis, P., and Dasmahapatra, S. (2010). A knowledgeable security model for distributed health information systems. *computers & security*, 29(3):331–349.

[397] Xu, D. and Zhang, Y. (2014). Specification and analysis of attribute-based access control policies: An overview. In *Software Security and Reliability-Companion (SERE-C), 2014 IEEE Eighth International Conference on*, pages 41–49. IEEE.

[398] Xuexiong, Y., Qinxian, W., and Changzheng, X. (2010). A multiple hierarchies rbac model. In *Communications and Mobile Computing (CMC), 2010 International Conference on*, volume 1, pages 56–60. IEEE.

[399] Yang, H. (2016). Cryptographic enforcement of attribute-based authentication. *Universitet i Agder*.

[400] Yang, N., Barringer, H., and Zhang, N. (2007). A purpose-based access control model. In *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on*, pages 143–148. IEEE.

[401] Yarmand, M. H., Sartipi, K., and Down, D. G. (2013). Behavior-based access control for distributed healthcare systems. *Journal of Computer Security*, 21(1):1–39.

[402] Yarmohammadian, M. H., Raeisi, A. R., Tavakoli, N., and Nansa, L. G. (2010). Medical record information disclosure laws and policies among selected countries; a comparative study. *Journal of research in medical sciences: the official journal of Isfahan University of Medical Sciences*, 15(3):140.

[403] Zambouri, A. (2007). Preoperative evaluation and preparation for anesthesia and surgery. *Hippokratia*, 11(1):13–21.

[404] Zamite, J., Domingos, D., Silva, M. J., and Santos, C. (2013). Group-based discretionary access control for epidemiological resources. *Procedia Technology*, 9:1149–1158.

[405] Zhang, R. and Liu, L. (2010). Security models and requirements for healthcare application clouds. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 268–275. IEEE.

[406] Zhao, L., Liu, S., Li, J., and Xu, H. (2010). A dynamic access control model based on trust. In *Environmental Science and Information Application Technology (ESIAT), 2010 International Conference on*, volume 1, pages 548–551. IEEE.

[407] Zhou, W. (2008). *Access control model and policies for collaborative environments*. PhD thesis, University of Potsdam.

[408] Zhou, W. and Meinel, C. (2007). Team and task based rbac access control model. In *Network Operations and Management Symposium, 2007. LANOMS 2007. Latin American*, pages 84–94. IEEE.

[409] Zhu, H., Lü, K., and Jin, R. (2009). A practical mandatory access control model for xml databases. *Information Sciences*, 179(8):1116–1133.

[410] Zinszer, K., Tamblyn, R., Bates, D. W., and Buckeridge, D. L. (2013). A qualitative study of health information technology in the canadian public health system. *BMC public health*, 13(1):509.